

**MATHIAS JOSÉ KREUTZ ERDTMANN**

**DESENVOLVIMENTO DE UMA PLATAFORMA ROBÓTICA  
MÓVEL INCLUINDO SISTEMA EMBARCADO DE VISÃO  
ESTÉREO**

**FLORIANÓPOLIS**

**2009**



**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS**

**DESENVOLVIMENTO DE UMA PLATAFORMA ROBÓTICA  
MÓVEL INCLUINDO SISTEMA EMBARCADO DE VISÃO  
ESTÉREO**

Dissertação submetida à  
Universidade Federal de Santa Catarina  
como parte dos requisitos para a  
obtenção do grau de Mestre em Engenharia de Automação e Sistemas.

**MATHIAS JOSÉ KREUTZ ERDTMANN**

Florianópolis, junho de 2009



# **DESENVOLVIMENTO DE UMA PLATAFORMA ROBÓTICA MÓVEL INCLUINDO SISTEMA EMBARCADO DE VISÃO ESTÉREO**

Mathias José Kreutz Erdtmann

'Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia de Automação e Sistemas, Área de Concentração em Controle, Automação e Sistemas, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina.'

---

Prof. Marcelo Ricardo Stemmer, Dr. Ing.  
Orientador

---

Prof. Eugênio de Bona Castelan Neto, Dr.  
Coordenador do Programa de Pós-Graduação em Engenharia de Automação e Sistemas

Banca Examinadora:

---

Prof. Marcelo Ricardo Stemmer, Dr. Ing.  
Presidente

---

Prof. Edson Roberto De Pieri, Dr.

---

Prof. Emerson Pereira Raposo, Dr.

---

Prof. Ubirajara Franco Moreno, Dr.



## **Agradecimentos**

À minha querida Cristiane, por todo afeto que me dedica, reciprocidade, atenção e amor.

Ao Professor Marcelo Ricardo Stemmer, pela confiança, orientação e todo conhecimento transferido ao longo de tantos anos de academia, que precedem a realização deste trabalho e que me renderam oportunidades únicas.

Aos meus pais, por seu apoio incondicional.

Aos colegas do grupo Robota, sejam os antigos com quem eu trabalhei diretamente (Dotto, Dank, Giuliano, Gustavo Freitas, Silvano) ou os novos (Clovis Scotti, Donadel, Eduardo Hülse, Lenhard, Roni), por promover o estudo da robótica móvel junto aos estudantes de graduação da(s) universidade(s).

Aos colegas do S2i, do passado remoto ao futuro recente. Vocês foram muitos e a convivência com cada um importante (Orth, Deschamps, Alberto, Roloff, Herdt, Terra, Grutz, Donada, Bess, Junges, Silvano, Rutzen, Musa, Clovis, Mauricio Stivanello, ...).

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes), por ter financiado este projeto pelos primeiros 18 meses de pesquisa.





Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para obtenção do grau de Mestre em Engenharia de Automação e Sistemas.

## **DESENVOLVIMENTO DE UMA PLATAFORMA ROBÓTICA MÓVEL INCLUINDO DE SISTEMA EMBARCADO DE VISÃO ESTÉREO**

**Mathias José Kreutz Erdtmann**

Junho/2009

Orientador: Marcelo Ricardo Stemmer, Dr. Ing.

Área de Concentração: Controle, Automação e Sistemas.

Palavras-chave: robótica móvel, sistemas de visão, visão estereoscópica

Número de Páginas: 140

O objetivo deste trabalho é desenvolver uma plataforma robótica móvel com sistema de visão estereoscópica embarcado. Para tanto, são avaliadas as diversas opções existentes no mercado de robótica móvel, optando-se por projetar e implementar toda a plataforma, obtendo assim uma maior flexibilidade no projeto. Esta flexibilidade permitiu a integração de sensores modernos de realimentação externa, como a bússola eletrônica, acelerômetros e sistema de visão monocular e estereoscópica. Foram implementados algoritmos para as funcionalidades básicas da plataforma, como o controle dos motores, navegação, mapeamento e localização, além de pesquisar-se mais profundamente a utilização de sistemas de visão monocular e estereoscópica para medição de perfis de distância. Foi também implementado o reconhecimento de objetos sintéticos para auxiliar a localização da plataforma. Como resultado, obteve-se uma plataforma completa e funcional, sobre a qual podem ser realizados testes variados sobre os tópicos que envolvem a robótica móvel e sistemas de visão embarcados.



Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Automation and Systems Engineering.

## **DEVELOPMENT OF A MOBILE ROBOT PLATFORM WITH EMBEDDED STEREO VISION SYSTEM**

**Mathias José Kreutz Erdtmann**

June/2009

Advisor: Marcelo Ricardo Stemmer, Dr. Ing.

Area of Concentration: Control, Automation and Systems.

Keywords: mobile robots, vision systems, stereo vision

Page count: 140

The purpose of this work is to develop a mobile robot platform with an embedded stereo vision system. In order to accomplish this, some of the available mobile robot platforms are evaluated, finally choosing the development of a whole new platform, with the main advantage of design flexibility. This flexibility allows the integration of modern sensor systems (including external feedback), such as electronic compass, accelerometers and stereo/mono vision systems. All the required algorithms for the basic functionalities were implemented, such as the motor control system, navigation, mapping and localization, and this work goes further into the issue of monocular and stereo vision for distance profile measurement. A system for localization using synthetic objects was implemented, too. The final result is a practical platform, over which it is possible to complete tests and researches about the several topics of mobile robots and embedded vision systems.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>17</b>
1.1	Objetivos . . . . .	19
1.2	Recursos disponíveis, trabalhos anteriores, grupos de pesquisa . . . . .	21
1.3	Organização do documento e convenções . . . . .	22
<b>2</b>	<b>Revisão do estado da arte em visão para robótica móvel</b>	<b>25</b>
2.1	Introdução . . . . .	25
2.2	Plataformas comerciais para robótica móvel . . . . .	26
2.3	Localização, mapeamento e navegação . . . . .	31
2.4	Visão no posicionamento . . . . .	38
2.4.1	Visão monocular . . . . .	39
2.4.2	Visão estereoscópica . . . . .	41
2.4.3	Visão híbrida . . . . .	44
2.4.4	Outras aplicações da visão computacional . . . . .	44
2.5	Considerações . . . . .	45
<b>3</b>	<b>Plataforma aberta para pesquisa em robótica móvel</b>	<b>47</b>
3.1	Concepção original . . . . .	47
3.2	Estrutura modular . . . . .	48
3.2.1	Opções para subsistema mecânico e interface . . . . .	50
3.2.2	Opções para subsistema eletro-eletrônico e interface . . . . .	52
3.2.3	Opções para subsistema computacional . . . . .	57
3.3	Análise e testes em protótipos . . . . .	58

---

3.3.1	Produtos disponíveis . . . . .	59
3.3.2	Protótipos anteriores . . . . .	60
3.3.3	Protótipo escolhido e modificado . . . . .	62
3.3.4	Projeto da plataforma atual . . . . .	65
3.3.5	Protótipos posteriores . . . . .	68
3.4	Considerações . . . . .	70
<b>4</b>	<b>Projeto e implementação dos subsistemas</b>	<b>71</b>
4.1	Subsistema mecânico . . . . .	71
4.1.1	Estrutura modular expansível . . . . .	72
4.1.2	Motores e reduções . . . . .	72
4.2	Subsistema eletro-eletrônico . . . . .	73
4.2.1	Armazenamento de energia . . . . .	73
4.2.2	Circuito de acionamento de potência . . . . .	74
4.2.3	Placa lógica e sensores . . . . .	75
4.2.4	Programação do microcontrolador . . . . .	81
4.3	Subsistema computacional . . . . .	82
4.3.1	Conexão USB . . . . .	83
4.3.2	Programação concorrente . . . . .	84
4.3.3	Mapeamento e planejamento de trajetória . . . . .	87
4.3.4	Modelagem cinemática e estratégia de controle . . . . .	89
4.3.5	Controle de ângulo . . . . .	93
4.3.6	Interação com usuário . . . . .	97
4.4	Sistema de Visão . . . . .	99
4.4.1	Visão monocular para medição de perfis de distância . . . . .	100

---

4.4.2	Visão estereoscópica . . . . .	103
4.4.3	Unificação das medições de distância . . . . .	106
4.4.4	Localização . . . . .	107
4.4.5	Integração com o Harpia . . . . .	109
4.5	Considerações . . . . .	111
<b>5</b>	<b>Resultados e Contribuições</b>	<b>113</b>
5.1	Projetos técnica e financeiramente viáveis . . . . .	113
5.2	Substituição de realimentação interna por realimentações externas . . .	114
5.3	Confiabilidade com sensores adicionais . . . . .	116
5.4	Facilidade para expansão e reprogramação . . . . .	116
5.5	Programação completa da estrutura básica para controle de alto nível da plataforma . . . . .	117
5.6	Visão (estereoscópica) embarcada com aplicações já testadas . . . . .	118
5.7	Flexibilidade para alterações drásticas, seguindo a mesma concepção bá- sica . . . . .	122
5.8	Considerações . . . . .	123
<b>6</b>	<b>Conclusão e perspectivas</b>	<b>125</b>
<b>A</b>	<b>Apêndice A</b>	<b>127</b>
<b>B</b>	<b>Apêndice B</b>	<b>129</b>
<b>C</b>	<b>Apêndice C</b>	<b>131</b>
	<b>Referências</b>	<b>133</b>





## 1. Introdução

O desenvolvimento da robótica móvel nas últimas décadas tem sido contido devido às dificuldades tecnológicas e, apesar das várias áreas necessárias terem sido desenvolvidas ao ponto que possam satisfazer as demandas das aplicações à robótica móvel, ainda não existe a integração necessária para tornar a aplicação da robótica móvel um ambiente conciso de trabalho, a exemplo do que houve após a formalização da computação, robótica de manipuladores e máquinas ferramenta.

Seria necessário, portanto, que um ambiente de integração fosse criado, a partir do qual seja possível a definição dos padrões que, por hora, são bem definidos na literatura e implementados como experiências acadêmicas pontuais e não facilmente reproduzíveis.

A área da simulação contribui consideravelmente para a popularização da robótica móvel, pois possibilita que testes sejam aplicados com poucos recursos físicos e de forma rápida. No entanto, apesar da crescente expansão do mercado de robótica móvel, ainda é necessário que se produza tanto equipamento (*hardware*) quanto a programação (*software*) que ultrapassem os limites da simulação a fim de criar produtos que incluam a possibilidade de utilização de robótica móvel em larga escala, sobretudo para inclusão de sistemas de visão. Assim, apesar da validade dos trabalhos de simulação, existe uma demanda no sentido de trabalhos verdadeiramente práticos e experimentais.

Sistemas de visão embarcados se tornaram viáveis nos últimos anos, em parte devido à miniaturização da eletrônica computacional e em parte à evolução dos algoritmos de visão de alto nível. Sistemas de visão baseados em técnicas bidimensionais geométricas já são amplamente utilizados industrialmente e embarcados mesmo em câmeras fotográficas comerciais de custo baixo, em forma de realçadores de imagem, detecção de faces para ajuste de foco, detecção das condições de iluminação e movimento. Uma nova gama de aplicações pode ser alcançada com o advento das técnicas de visão estereoscópica, possibilitando a extração de informação tridimensional seguindo o modelo antropomórfico de visão.

Dentro deste cenário que envolve tecnologias em ascensão e desafios tanto da ordem operacional quanto da ordem científica se insere este trabalho, que se propõe a desenvolver uma plataforma sobre a qual possam ser projetados sistemas robóticos móveis. Assim, é feita uma separação dos subsistemas inclusos em um sistema robótico móvel e são analisadas algumas opções de projeto de cada subsistema. Ainda dentro desta análise, são avaliados protótipos incluindo algumas das combinações possíveis, no sentido de realizar breves estudos de caso.

Com base nos resultados obtidos com os protótipos, propõe-se o projeto de uma plataforma robótica móvel, levando em conta requisitos de flexibilidade, expansibilidade e custo. Adicionalmente, um requisito primordial é a possibilidade de realizar estudos acadêmicos na área de visão computacional, em especial visão estereoscópica, o que torna o projeto mais complexo e mais interessante.

Dada a necessidade de tal projeto utilizar princípios de várias áreas da engenharia, como mecânica, elétrica e computação, adotou-se a partir do princípio a regra de modularidade e simplicidade, sem as quais seria impossível progredir em um projeto amplo como o proposto. Assim, por vezes a utilização de soluções simplistas envolve, sobretudo, a verificação de adequação para que seja possível obter bons resultados (tanto no teste da solução isolada quanto num âmbito global), não obstante as simplificações exigidas pelo projeto, caracterizando assim o trabalho do ponto de vista da engenharia com o compromisso entre viabilidade e desempenho.

Ainda assim, do ponto de vista científico, existem os problemas interessantes que emergem da interação da robótica móvel e sistemas de visão, que por um lado fornecem flexibilidade e por outro, uma quantia grande de dados a serem processados em um intervalo de tempo pequeno. No entanto, mesmo que o tempo disponível para processamento seja restrito, do ponto de vista de taxa de atualização de sensores, o tempo é muito grande, fornecendo pouca informação por unidade de tempo, quando comparado com sensores eletrônicos tradicionais. Resumindo, o problema científico posto é obter o maior número de informações úteis e confiáveis no menor tempo possível, a partir dos dados das imagens. Este problema é um problema atual, salientado por vários trabalhos como a execução de processamento de imagens em tempo real para robótica

[36, 49, 71, 44]. Como se pode imaginar, este problema é bastante agravado quando se trata de visão estereoscópica, pois a quantidade de dados a ser processados dobra com a utilização de duas câmeras e a correlação entre os dados da imagem de uma câmera com a da outra é um problema ainda em aberto.

Por último, tecnologicamente, o trabalho se insere em um momento onde a indústria eletrônica reforça a área de sensoriamento e comunicação, mantendo ainda o ritmo de crescimento exponencial da capacidade de processamento. Assim, é possível agora combinar sensores que medem variáveis ambientais externas ao robô (fluxo magnético, aceleração), poder de processamento para reunir todos os dados e comunicação sem fio confiável com a qual é possível a interação, depuração e acompanhamento do usuário de maneira rápida e eficiente. Neste sentido, o maior trabalho do ponto de vista tecnológico reside na pesquisa, análise e integração das diversas soluções disponíveis.

## 1.1: Objetivos

Baseado no contexto especificado na seção anterior resume-se o objetivo geral do trabalho em:

**Desenvolver uma plataforma robótica móvel com sistema de visão estereoscópica embarcado.**

Mais especificamente, deseja-se alcançar ainda os seguintes objetivos:

- Analisar protótipos construídos, realizando novos testes e realizar o novo projeto de forma a satisfazer requisitos de comunicação e desempenho dinâmico.
- Analisar as possibilidades de aplicações com sistemas de visão embarcados em robótica móvel, como: detecção e reconhecimento de objetos e faces, tarefas de vigilância.
- Estudar as capacidades das opções de produtos atualmente disponíveis na área de sensoriamento.
- Criar uma interface entre homem e máquina via rede sem fio, seguindo padrões de comunicação vigentes.

- Permitir a flexibilidade da plataforma, de forma que ela possa ser rapidamente utilizada para testar aplicações em visão estereoscópica e facilmente modificada para outras aplicações.
- Desenvolver um sistema de localização, mapeamento e navegação baseado na realimentação visual.
- Utilizar visão estereoscópica de bom desempenho para controle de trajetória, modificando as bibliotecas disponíveis atualmente, se necessário.
- Disponibilizar os projetos de equipamento e programas de forma livre.

Como limitação do escopo do trabalho, se propõe que a plataforma seja constituída de um robô portátil (da ordem de 30cm em profundidade, largura e altura), capaz de suportar e operar com 10kg de carga. Deve haver espaço suficiente para montar sobre ele sistemas manipuladores, sistemas de visão com câmeras maiores, módulos extra de processamento de sinais, prevendo-se pelo menos um espaço de 30x30cm de superfície superior e estabilidade para uma altura de 60cm.

O ambiente no qual a plataforma deve ser capaz de operar se constitui um ambiente semi-estruturado, caracterizado por ambientes fechados, os quais pessoas normalmente ocupam, mas livre de obstáculos, tais como degraus com mais de 3cm de altura, pedras soltas. Um exemplo desses ambientes são os escritórios, podendo conter cadeiras, bolsas no chão, livros, variações de padrões de divisória, paredes, móveis (armários, gaveteiros), no entanto, sempre possuindo passagens livres.

Destaca-se que o objetivo é a construção da plataforma, e não necessariamente um robô completo com atuadores além dos necessários para sua própria locomoção e funcionamento. Ainda assim, o projeto deve se adequar prevendo-se o futuro desenvolvimento de manipuladores, ou mesmo a aplicação para tarefas de inspeção sem atuação ativa em objetos do ambiente.

## 1.2: Recursos disponíveis, trabalhos anteriores, grupos de pesquisa

A pesquisa foi realizada nas dependências do LTIC e contou com o apoio do grupo de pesquisa S2i - Sistemas Industriais Inteligentes e pelo grupo de estudos de robótica móvel – Robota. Além do equipamento típico do ambiente de escritório, como computadores, impressoras, o LTIC e seus diversos grupos de estudos contam com as ferramentas necessárias para projetos e implementações de sistemas eletrônicos e computacionais [21].

Especificamente, o S2i conta com o material necessário para projeto e implementação de sistemas de visão e sistemas industriais, incluindo câmeras, lentes, sistemas de montagem de bancadas, sistemas de iluminação, gravadores de microcontroladores, computador e sensores industriais, e uma extensa biblioteca de processamento de imagens desenvolvida no decorrer de cerca de uma década de trabalho, incluindo os trabalhos recentes de visão estereoscópica realizados por Maurício Stivanello [61, 62].

Para o trabalho de projeto e implementação das estruturas mecânicas e o trabalho de programação computacional, o projeto contou com a colaboração dos integrantes do grupo Robota ao longo do período de trabalho (2007: Clovis Peruchi Scotti, Eduardo Otte Hülse, Renata Faraco Cunha, Tiê Teixeira Lima Penatti; 2008: Clovis Peruchi Scotti, Roni Rigoni, Rodrigo Donadel, Gabriel Felipe Lehnhard), utilizando-se ainda dos conhecimentos prévios e trabalhos anteriores do grupo com os trabalhos desenvolvidos entre os anos de 2005 e 2008 (o Apêndice A apresenta uma lista de integrantes dos projetos que compuseram o conjunto de protótipos avaliados neste trabalho).

Cabe salientar que os protótipos apresentados pelo grupo Robota tipicamente haviam sido projetados com o intuito de participar nos eventos do LARC (*Latin American Robotics Competition*) entre 2005 e 2008. A competição anual, promovida pelo IEEE (*Institute of Electrical and Electronics Engineers*), visa promover o desenvolvimento do estudo da robótica móvel autônoma em grupos de graduação e pós-graduação na América latina e conta com um número crescente de equipes participantes. O instituto vem ressaltando com a competição a tendência de desestruturação do ambiente de navegação e necessidade dos sistemas de visão embarcados, propondo desafios de crescente dificuldade abrangendo estes problemas [20].

Os trabalhos anteriores que permitiram a elaboração deste, de ambos os grupos Robota e S2i, contaram com a participação do autor como membro das equipes de desenvolvimento, e, no trabalho aqui apresentado, a contribuição pessoal do autor é relacionada aos seguintes tópicos:

- Projeto mecânico, fabricação e montagem das peças;
- Projeto eletro-eletrônico, formalização dos projetos anteriores, confecção das diversas placas, seleção dos componentes.
- Projeto computacional, toda a programação realizada (microcontrolador e IBM/PC), modelagem e testes, com exceção da visão estéreooscópica, adaptada do trabalho de Mauricio Stivanello.

Destaca-se que tanto o projeto mecânico quanto eletro-eletrônico utilizam as concepções anteriores dos grupos, no entanto foram realizados como trabalhos individuais do autor.

Além dos trabalhos do grupo Robota e S2i, o Departamento de Automação e Sistemas conta com diferentes grupos de pesquisa, abordando diversos aspectos da robótica móvel. Assim, podem ser destacados vários trabalhos não diretamente correlacionados com este, mas que participam do contexto da pesquisa no departamento, tais como trabalhos de modelagem e controle [48] e de robôs bípedes [32].

As matérias primas utilizadas na fabricação dos protótipos e da plataforma foram fornecidas por uma grande variedade de fontes, incluindo o Departamento de Automação e Sistemas e seus diversos professores, os laboratórios de fabricação do Departamento de Engenharia Mecânica (usinagem, soldagem e conformação) e doações particulares.

### **1.3: Organização do documento e convenções**

No capítulo 2 deste documento será apresentada uma revisão do estado atual dos estudos em robótica móvel, sobretudo no que diz respeito aos sistemas de navegação e mapeamento, sistemas de visão e como tais pesquisas se relacionam. São

apresentadas ainda as várias opções disponíveis no mercado, seguidas de uma breve análise das opções de projeto adotadas pelos fabricantes.

O capítulo 3 descreve a plataforma concebida do ponto de vista conceitual. Nesta descrição geral é apontado o processo de projeto da plataforma, partindo da concepção original, a criação de uma estrutura modular e testes de protótipos. Os protótipos anteriores são analisados tendo em vista as várias escolhas possíveis de elementos dentro da estrutura descrita, assim ilustrando o processo de decisão da solução proposta e das melhorias previstas.

A partir da escolha da solução a ser utilizada para cada módulo, e baseando-se nas funcionalidades previstas na concepção, passa-se aos projetos detalhados dos vários subsistemas que compõe a plataforma, que são descritos no capítulo 4. Neste capítulo detalha-se a implementação dos subsistemas mecânico, eletro-eletrônico e computacional. Dentro do subsistema computacional incluem-se os trabalhos realizados com visão computacional.

O capítulo 5 integra os vários sistemas, resumindo os resultados alcançados e incluindo resultados obtidos a partir da plataforma final, com todos os sistemas integrados e funcionais.

Por fim, o capítulo 6 apresenta as considerações finais, incluindo uma breve análise do que foi alcançado com este trabalho e algumas possibilidades para estudos futuros.

Muito embora o robô seja constituído da plataforma robótica desenvolvida em conjunto com os demais sistemas embarcados nela, como atuadores, ao decorrer do documento, por simplicidade, serão adotados como sinônimos: plataforma, plataforma robótica e robô.





## 2. Revisão do estado da arte em visão para robótica móvel

### 2.1: Introdução

Os primeiros trabalhos no sentido de se utilizar visão computacional para mapeamento e navegação mostram as dificuldades tecnológicas enfrentadas, como em especial, a visão estereoscópica em [45]. Eram necessários o processamento remoto e também unidades de processamento de sinais implementadas especificamente para a aplicação.

A partir do ano 2000, no entanto, as dificuldades quanto ao poder de processamento foram reduzidas devido ao avanço dos computadores, permitindo que aplicações distintas aparecessem como experiências separadas, como o sistema de navegação noturno proposto em [52] e os diversos trabalhos que fazem parte desta revisão.

Em uma tentativa de unificar a teoria acerca de sistemas de visão e navegação de robôs, foi publicado o artigo *"Vision for Mobile Robot Navigation: A Survey"* [31], separando as técnicas em duas categorias: navegação interna e externa. Na navegação interna, os tópicos de interesse são o mapeamento e detecção de características importantes, enquanto na navegação externa diminuem-se os detalhes e busca-se seguir estradas e detectar obstáculos em ambientes não estruturados. Como resultado, os autores do artigo estimam que as tarefas simples de movimentação poderiam ser realizadas, mas tarefas específicas, como parar em placas **PARE** e buscar extintores de incêndio ainda seriam um desafio para o futuro, portanto se tratando de problemas interessantes e que exigem soluções inovadoras, pois em geral não são ainda apresentadas soluções para elas.

Os resultados do processo de sistematização dos conceitos de robótica móvel podem ser vistos em [59], que abrange os tópicos de modelagem da dinâmica do robô, mapeamento, navegação e uma série de outros tópicos, não cobrindo especificamente a parte de visão computacional, mas implementação de robôs móveis em geral (com ênfase em robôs com rodas). Deste processo de sistematização pode-se prover várias soluções para os problemas básicos de movimentação de robôs [58].

No campo de visão estereoscópica, existe muita pesquisa na área de base, buscando algoritmos velozes e com qualidade cada vez maior, como o trabalho de Maurício Edgar Stivanello usado como base para este trabalho [61, 62, 28].

Assumindo que os algoritmos de visão estereoscópica já possuam desempenho satisfatório, as aplicações (em robótica) se mostram variadas, podendo ser divididas em algumas classes:

- Detecção de obstáculos [29, 64, 44, 68]
- Reconhecimento de regiões características (p.ex. caminho, árvores) [53]
- Reconhecimento de objetos (p.ex. face, peças) [39, 49]
- Navegação e/ou mapeamento [42, 35]

Com a crescente aplicação e os bons resultados obtidos, iniciativas no sentido de tornar a aplicabilidade da visão estereoscópica viável do ponto de vista comercial começam a surgir, como, por exemplo, o sensor estereoscópico dedicado visto em [71] e o sistema de frenagem automática para manutenção da distância de segurança entre dois veículos descrito em [66].

## **2.2: Plataformas comerciais para robótica móvel**

Apesar da ampla utilização de robôs manipuladores, sobretudo no ramo industrial, estes sofrem de uma limitação de mobilidade que impede que muitas tarefas possam ser executadas por este tipo de robô. No entanto, ao manter a base do robô fixa, assume-se que o ambiente não será alterado, simplificando o problema e tornando possível a aplicação de robôs com a confiabilidade exigida do ambiente industrial.

As dificuldades em criar um robô móvel autônomo residem justamente no fato de que, por ser móvel, o ambiente ao redor do robô se altera, fazendo com que seja necessário um novo nível de flexibilidade e processamento de dados para que o robô se adapte a situação atual no qual se encontra [59].

Devido às limitações dos sensores e algoritmos atuais de processamento, ocorrem muitas situações onde ambientes distintos passam a ser indistinguíveis uns dos

outros. Assim, distinguir ambientes diferentes e identificar a ocorrência do mesmo ambiente previamente visitado tornam-se problemas de solução bastante sutil, cuja sensibilidade torna-se crítica a medida que se use sensores mais inexatos.

Assim, a produção de robôs autônomos tem sido bastante contida, enquanto máquinas tele-operadas já vêm sendo usadas amplamente, devido à maior confiabilidade do operador humano [23].

Para o ramo acadêmico, já existem alguns robôs móveis autônomos disponíveis, enquanto comercialmente estes produtos ainda não podem ser considerados maduros, a não ser em algumas aplicações simples e casos pontuais. A seguir será feita uma breve descrição de alguns sistemas disponíveis no mercado (o Apêndice B apresenta uma tabela comparativa destes sistemas).

O líder no segmento acadêmico é o robô Pioneer, da MobileRobots Inc. (Figura 1). O robô Pioneer apresenta uma arquitetura modular e expansível, de forma que estudos bem variados foram gerados com a utilização desta plataforma. No entanto, enquanto expansões possam ser realizadas, a arquitetura fechada e os custos elevados das expansões fabricadas pela empresa dificultam o estudo do processo de projeto e construção de robôs propriamente ditos, enquanto por outro lado facilitaram a pesquisa de algoritmos para o equipamento existente. A empresa desenvolve ainda outros modelos de robôs móveis de variados tamanhos e equipamentos auxiliares [12].



Figura 1: Plataforma Pioneer, da MobileRobots Inc. [12]

A empresa White Box Robotics desenvolveu um robô móvel com arquitetura aberta chamado PC-BOT (Figura 2), a idéia original da empresa é de seguir a modularidade

e capacidade de atualização de equipamentos como vistos nos computadores pessoais IBM/PC. A arquitetura aberta facilita com que o robô possa ser personalizado de acordo com as necessidades do usuário [17].



Figura 2: Robô PC-BOT 914, da White Box Robotics [17]

O robô Pekeell, da Wany Robotics (Figura 3), lançado em setembro de 2008, disponibiliza um conjunto amplo de expansões mostrando o sucesso da arquitetura modular, em um projeto compacto e robusto. Sensores modernos, como bússola e acelerômetros estão disponíveis entre os opcionais. A arquitetura computacional com API disponível em várias linguagens de programação e compatibilidade com vários programas de computação científica auxilia a pesquisa utilizando esta plataforma. A arquitetura fechada, no entanto, tende a dificultar modificações além dos módulos disponíveis pela companhia [16].

O robô SRV-1 Blackfin (Figura 4), da Surveyor Corporation, utiliza um microcontrolador de alto desempenho para unificar o sensoriamento, sistema eletro/eletrônico e computacional, incluindo câmera e comunicação sem fio. Trata-se de um robô de pequeno porte com várias funcionalidades e expansível por meio da placa de controle inclusa, em um projeto aberto (tanto a parte computacional quanto eletrônica). Apesar do baixo custo, este robô apresenta limitações físicas quanto à possibilidade de expansão



Figura 3: Robô Pekeell, da Wany Robotics [16]

e computacionais devido aos recursos escassos de processamento e armazenamento de dados da placa microcontroladora [15].

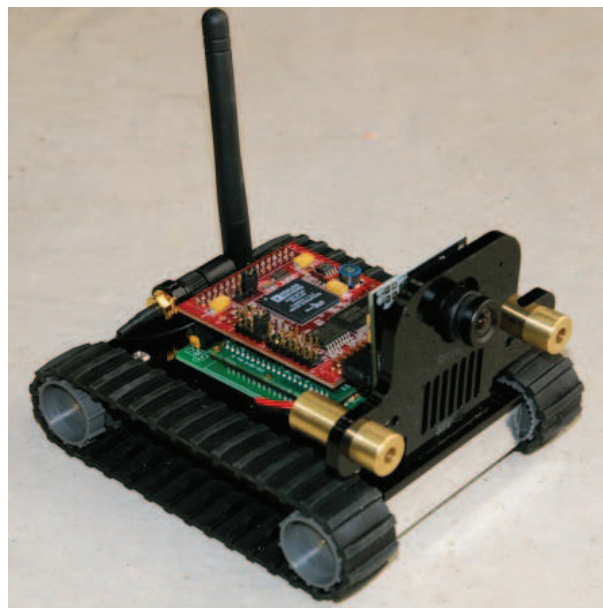


Figura 4: Robô SRV-1 Blackfin, da Surveyor Corporation [15]

No ramo comercial/doméstico, robôs para limpeza (aspiradores de pó) já podem ser encontrados, como o Roomba (iRobot Corporation), e o Cleanmate QQ-2(Infinuvo Corporation) (Figura 5). Estes robôs apresentam a funcionalidade de percorrer as áreas desviando de obstáculos, executando trajetos que se espera que recubram toda a área a ser limpa, no entanto nenhum algoritmo mais complexo de mapeamento é executado [9, 8].



Figura 5: Robô Roomba (esq.) e Cleanmate QQ-2 (dir.) [9, 8]

No que se refere a robôs com pernas, os estudos são ainda mais recentes e poucos são os produtos já disponíveis para este tipo de robô móvel. Em especial, alguns robôs bípedes podem ser encontrados, como por exemplo, o pequeno Robonova (da Hitec Inc.) e o já famoso ASIMO(Honda Motor Co.) (Figura 6). Destes dois, somente o Robonova está disponível no mercado [7, 3].



Figura 6: Robôs bípedes Robonova (esq.) e ASIMO (dir.) [7, 3]

Assim, percebe-se que existe um esforço crescente para a pesquisa em robótica móvel, e o sucesso de mercado de produtos como os aspiradores de pó robótico mostram que o público doméstico também se interessa pelos produtos da robótica móvel. Foram apresentadas somente uma parcela das plataformas disponíveis, sobretudo as

que ofereciam soluções completas e generalizadas com processamento de sinais dos sensores e uma base para programação computacional.

No sentido em que facilita o desenvolvimento, as plataformas abertas são muito mais interessantes academicamente, pois permitem uma maior flexibilidade que é, de fato, essencial para uma linha de pesquisa tão nova quanto a robótica móvel. Essa tendência pode ser acompanhada pelos produtos ofertados pelas empresas na linha acadêmica, que buscam cada vez mais interoperabilidade, modularidade e abertura das especificações, ou, mais além, abertura da implementação computacional e eletrônica.

Além das soluções apresentadas, outras opções são disponíveis, por exemplo como bases holomônicas [4], bases para a parte mecânica da plataforma [10], e robôs para propósitos específicos, como a competição Robocup [18, 60]. Bases holomônicas permitem que os movimentos de translação e rotação da base sejam controlados independentemente, no entanto ainda são apresentados como produtos para o mercado de montagem de robôs móveis, ainda não constituindo sistemas de propósito geral como os aqui analisados.

### **2.3: Localização, mapeamento e navegação**

A questão da localização, mapeamento e navegação, foi um dos primeiros problemas a serem propostos no histórico da robótica móvel, sobretudo porque o caráter móvel acaba atrelando a idéia de mudar de posição e local, o que faz com que seja necessária alguma representação dos lugares e alguma estratégia para, dada a posição atual, chegar a algum lugar desejado [31].

Abordagens mais simples baseiam-se na ideia de apenas desviar de obstáculos, permitindo assim o robô se locomover sem colidir, sem, no entanto dar uma idéia precisa de onde está e para onde vai. Esta abordagem apesar de limitada, pode ser útil devido à simplicidade de implementação e por não exigir muito sensoriamento, pois é possível desviar de obstáculos sem muita informação sobre o mundo externo, apenas sabendo se há ou não algo à frente. Um exemplo de produto que utiliza esta abordagem são os aspiradores de pó robóticos domésticos citados anteriormente [70].



No entanto, para tornar o problema mais interessante, deseja-se sempre que o robô seja capaz de estimar sua posição (localização), conhecer o ambiente (mapeamento) e definir como chegar a algum lugar determinado (navegação).

### Mapeamento

A primeira coisa a definir quando se deseja implementar o sistema de posicionamento é o modelo de mapa, pois a partir deste modelo serão criadas as estruturas de dados que serão implementados todos os algoritmos para o funcionamento.

Os principais modelos de mapa encontrados na literatura são: mapa em grade discreta, mapa vetorial, mapa topológico (Figura 7) [70].



Figura 7: Representação gráfica dos modelos de mapa (posição do robô em vermelho e obstáculos azuis)

O mapa em grade discreta consiste em dividir o espaço em várias células, onde é possível marcar uma célula como ocupada por um obstáculo ou não. O robô pode estar e passar somente em células que não estejam ocupadas por obstáculos.

A posição do robô neste modelo é dada pela posição de uma das células e sua orientação. Em geral o espaço utilizado na subdivisão é o cartesiano 2D, representativo do chão sobre o qual o robô se movimenta, no entanto existem versões espaciais (tridimensionais) ou mesmo representações em outros espaços, que podem ser espaços representativos apenas para o robô em função de seus sensores. Este tipo de representações internas baseadas em sensores tem a desvantagem de ser de difícil compreensão para o usuário, mas podem ajudar a resolver problemas onde os lugares são muito parecidos entre si (problema do *aliasing*) [59].

Existem algumas variações do mapa em grade que permitem armazenar outros



dados nas células, como, por exemplo, o grau de certeza da ocupação da célula [70].

Como a grade tem tamanho conhecido e em geral fixo, pode-se estimar tempo máximo necessário para executar os algoritmos que, além disso, tendem a ser mais velozes que nos outros modelos. No entanto, a grade de tamanho fixo também implica em uma exatidão máxima equivalente à resolução do mapa (que é limitado pelo tempo de processamento, que em geral cresce quadraticamente com o número de divisões da grade).

Mapas vetoriais armazenam os obstáculos como uma lista de entidades geométricas (pontos, retas ou polígonos, entre outros) dentro do espaço. Neste caso, são afetados apenas pelas inexatidões das medições dos sensores usados para inserir as informações de posição dos obstáculos. Nestes mapas a posição do robô é representada como um ponto no espaço, junto com um vetor unitário indicando a orientação do robô.

No entanto, mapas vetoriais têm a desvantagem dos algoritmos para planejamento tenderem a aumentar seu custo à medida que novos obstáculos sejam acrescentados, de forma que é comum se discretizar mapas vetoriais a fim de realizar a navegação. [70]

Por último, mapas topológicos armazenam a relação de vizinhança entre zonas em forma de grafo, onde cada zona é um nó e cada passagem uma aresta cujo peso é a distância entre as zonas. Nota-se que os obstáculos são implícitos, uma vez que o que é destacado pelas arestas são as passagens. Mapas topológicos tem as vantagens de armazenar informações em uma estrutura leve (geralmente se tem poucos nós) e permitir que algoritmos de busca e roteamento conhecidos sejam aplicados (por exemplo o algoritmo de Dijkstra), no entanto, existe uma dificuldade operacional em transformar as rotas em caminhos físicos e reconhecer a posição atual como um nó atual do grafo, para evitar que o mesmo lugar possua dois nós no grafo. [70]

Desta forma, mapas topológicos muitas vezes acabam utilizando mapas em grade em pequena escala para resolver problemas locais e a topologia para resolver problemas em um nível mais alto.

Pode-se observar uma relação entre a grade e o modelo topológico, onde cada célula da grade seria um nó ligado às células adjacentes se elas estiverem livres por uma aresta de peso unitário (Figura 8). Podem-se usar algoritmos de grafos para simplificar o grafo obtido da grade para gerar um grafo topológico simplificado equivalente. Baseadas nesta ideia, existem abordagens mistas que tentam aproveitar as vantagens de cada método.

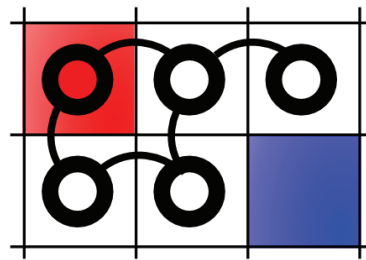


Figura 8: Relação entre modelo de grade e topológico

### Navegação

Dado o modelo do mapa, o processo de navegação consiste em achar um caminho entre a posição atual e a posição final desejada (planejamento) e manter a trajetória desejada ou alterá-la caso necessário (controle de trajetória). Cada modelo de mapa tem algoritmos específicos para cumprir o requisito de planejamento de trajetória, enquanto o controle é semelhante para qualquer planejamento, pois consiste no comando dos motores e re-planejamento quando necessário (quando existir informações novas).

A navegação em mapas de grade discreta pode ser resolvida, por exemplo, usando métodos de campos potencial e otimização (Figura 9)[59].

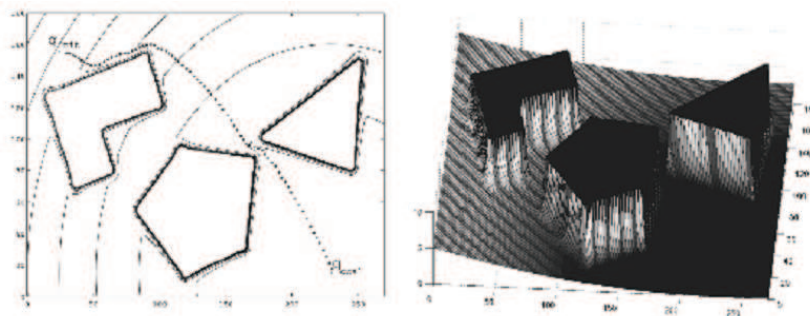


Figura 9: Método do potencial [59]

O inconveniente de métodos de otimização absoluta é que eles podem ser bastante demorados para grades com muitas células, o que fez com que surgissem métodos que encontrassem um caminho sub-ótimo mais rapidamente utilizando, por exemplo, métodos de inteligência artificial (Figura 10)[34]. Em geral, soluções sub-ótimas são satisfatórias, pois o principal objetivo é encontrar algum caminho, sempre que ele existir, ou chegar a conclusão que de fato não existe caminho – se este objetivo for cumprido, o problema já é considerado resolvido.

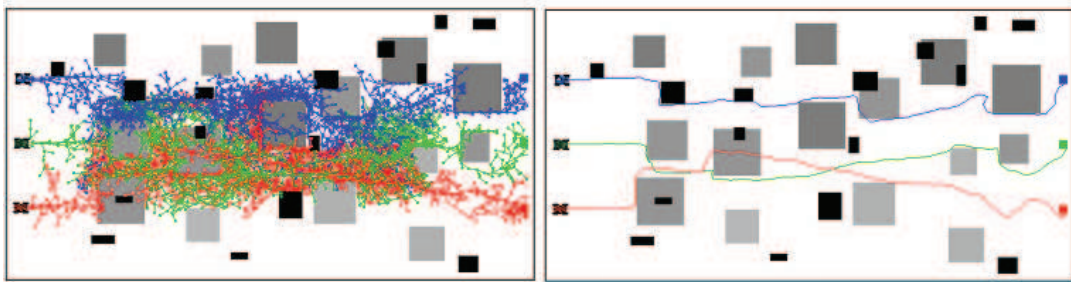


Figura 10: Método sub-ótimos. Árvore de busca (esq.) e caminho encontrado (dir.)[34]

Mapas vetoriais tipicamente são transformados em grade (discretizados) para então utilizar os métodos anteriores para planejamento da trajetória, no entanto existem ainda técnicas geométricas que utilizam diretamente o mapa vetorial, conforme mostra a Figura 11 [59].

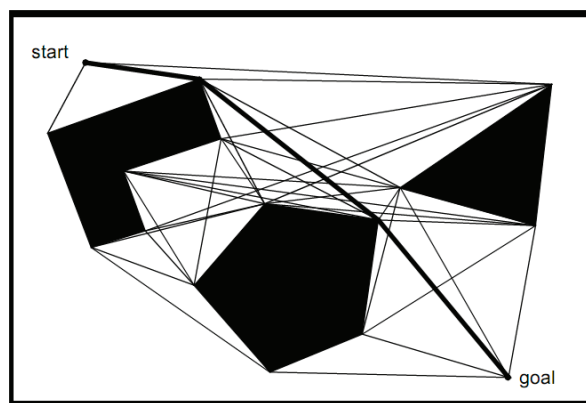


Figura 11: Método vetoriais diretos [59]

Mapas topológicos podem usar algoritmos de grafos como o algoritmo de Dijkstra ou  $A^*$  para calcular o caminho ótimo levando em consideração o peso das arestas. Caso

se deseje simplesmente achar um caminho qualquer, pode-se ignorar o peso e utilizar algoritmos mais leves, como a busca em profundidade (do inglês *Depth first search*). A Figura 12 mostra um exemplo de grafo com caminho encontrado entre dois nós.

Esses mesmos algoritmos podem ser usados quando se tratar da representação em grade via grafos, conforme ilustrado anteriormente.

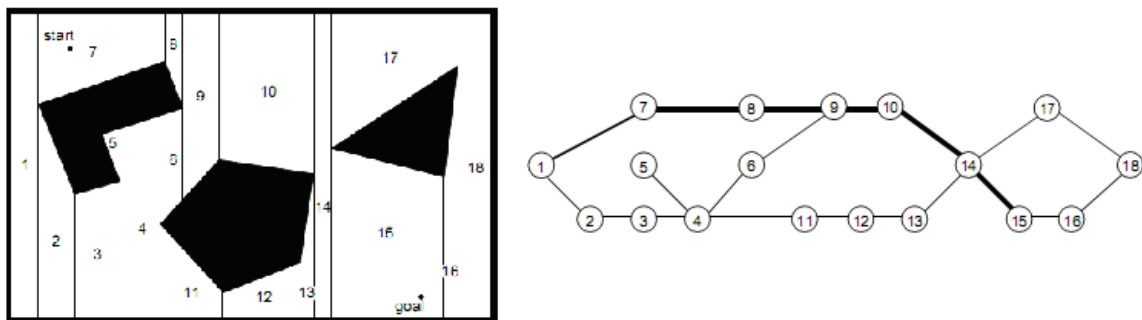


Figura 12: Método topológicos por grafos [59]

É importante notar que, além dos sistemas de navegação, as abordagens mais recentes incluem também o comportamento reativo, que reage à situações inesperadas, tais como obstáculos desconhecidos, assim constituindo uma navegação baseada em mapas combinados com sistemas reativos.

### Localização

Em geral, assume-se que todo movimento do robô se dá por comandos executados pelo próprio robô, de forma que a primeira idéia para localização é utilizar os dados de movimento esperado do robô dado o comando executado para atualizar a posição atual (caso o robô seja comandado a ir para frente com uma dada velocidade por um certo tempo, espera-se que sua posição final esteja uma certa quantia para frente). Esta técnica se chama integração de caminho (em inglês *dead reckoning*) e fornece uma estimativa da posição total percorrida utilizando um modelo dos motores e o sinal de controle fornecido.

Uma grande limitação deste método é que o erro tende a crescer rapidamente devido aos erros do modelo e efeitos adversos que fazem com que o comando executado não tenha exatamente o efeito modelado. Como a técnica é integrativa (soma incremen-

talmente distâncias percorridas) os erros somam-se e crescem indefinidamente. Este efeito é muito agravado em erros de orientação, pois um erro de poucos graus pode gerar um erro da ordem de metros a uma distância suficiente (Figura 13).

Com o auxílio de sensores pode-se reduzir bastante o erro, como por exemplo, bússola para limitar o erro de orientação (o erro não é mais integrado, fica limitado pelo erro de medição da bússola apenas) e *encoders* nas rodas para eliminar erro de modelagem do motor (mantém ainda erros de modelagem entre roda e chão). No entanto, mesmo com sensores, o crescimento indefinido do erro faz com que o uso somente desta técnica seja inviável em longo prazo.

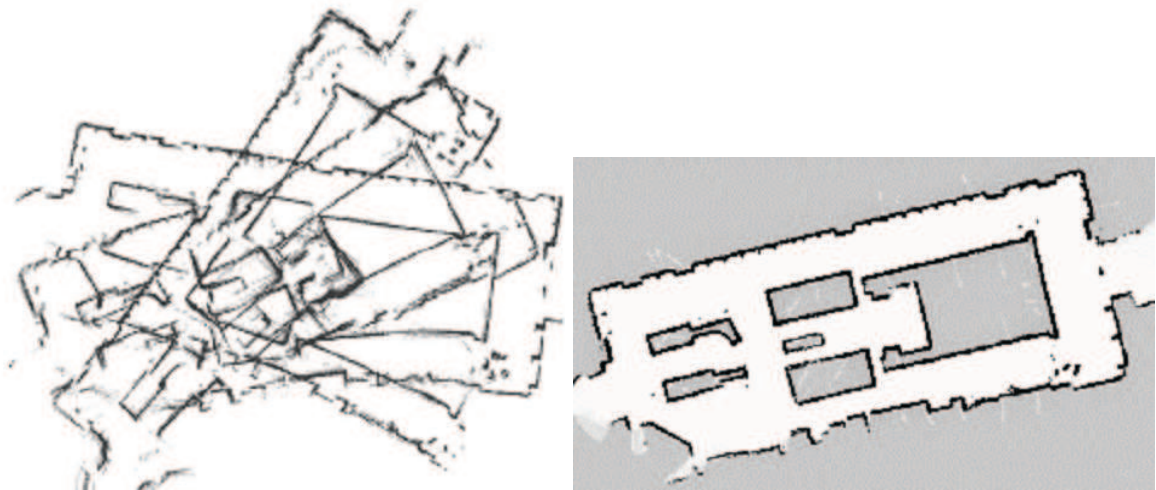


Figura 13: Mapa com problemas devido aos erros de medição de rotação (esq.) e com rotação corrigida (dir.)[59]

Ao conhecer a posição de objetos fixos no espaço, o robô pode, ao detectá-los, corrigir sua estimativa de posição baseada na posição do objeto com relação a ele. O uso desta técnica se chama localização baseada em marcos (do inglês *landmarks*). Esta abordagem é interessante, pois permite que erros gerados na integração de caminho sejam eliminados.

Extrapolando a abordagem dos marcos, trabalhos mais recentes buscam utilizar toda entrada de sensores para atualizar a posição, comparando as leituras atuais com as leituras marcadas no mapa e, ao parear a leitura com uma posição anteriormente armazenada, atualizar a posição atual do robô, assumindo que ele se encontra na posi-

ção de melhor pareamento<sup>1</sup>. Esta abordagem pode incluir cálculos estatísticos sobre a incerteza da posição e um conjunto de posições factíveis [70].

A tendência atual é unificar os processos de localização e mapeamento, no chamado SLAM (do inglês *simultaneous localization and mapping* – localização e mapeamento simultâneos). Neste caso, a localização e mapeamento são considerados processos contínuos e mesmo quando o robô deseja apenas navegar, em geral é possível e desejável manter a atualização da posição estimada e do mapa [43].

Com relação ao SLAM, grandes sucessos na integração de medições de sensores e variáveis previstas por modelos internos (como integração de caminho) têm sido alcançados com a aplicação do filtro de Kalman nas diversas variáveis envolvidas [70].

## 2.4: Visão no posicionamento

Câmeras são sensores funcionalmente flexíveis, mas em geral provêem dados em excesso, que precisam ser processados, filtrados e organizados na forma de informação útil. Uma das possibilidades de uso de câmeras é a medição de distâncias, eventualmente substituindo sistemas de medição via sonar ou laser nas tarefas de navegação dos robôs móveis. A medição de distâncias é o ponto inicial para as tarefas de mapeamento, pois é a partir das distâncias até os objetos que, dada a posição atual do robô, os obstáculos são acrescentados no mapa. Tipicamente três diferentes abordagens podem ser utilizadas:

- Abordagem monocular: usando uma câmera, assumindo certas condições, a região sobre a qual o robô navega (chão) é segmentada e as distâncias até os objetos que se encontram sobre o chão podem ser medidas, gerando um perfil de distância dos objetos em frente à câmera.
- Abordagem estereoscópica: usando duas câmeras alinhadas e semelhantes, um mapa de profundidades é gerado, e a informação do mapa de profundidade é utilizada para gerar o perfil de distância.

---

<sup>1</sup>do inglês *matching*, traduzido algumas vezes como casamento

- Abordagem híbrida: Os perfis gerados pelos dois métodos anteriores são utilizados no sentido de obter um perfil de distância mais completo.

Cada abordagem apresenta resultados diferentes e complexidade distinta, de forma que a aplicação de uma dada abordagem depende, na verdade, das necessidades da aplicação. A seguir cada abordagem será analisada e são fornecidos alguns exemplos de situações aonde cada uma vem sendo utilizada.

Além destas classificação, todas as abordagens podem incluir o uso de visão ativa, que possibilita o sistema de visão utilizar projetores digitais ou lasers, comandados pelo próprio sistema de visão de forma a ressaltar características para uma medição mais fácil e mais exata [27]. Estas abordagens têm obtido bons resultados, no entanto devido à necessidade de mais equipamento e testes, o estudo e implementação deste tipo de sistema de visão não foi realizado neste trabalho.

#### **2.4.1: Visão monocular**

Conforme proposto em [55], [54], [30] and [40], a informação sobre profundidade pode ser coletada utilizando apenas imagens únicas, sendo então essa informação utilizada para alimentar um sistema de navegação para um robô móvel autônomo. Como os testes são normalmente realizados em ambientes estruturados ou semi-estruturados, em geral assume-se que os obstáculos têm contato com o solo.

O algoritmo geral consiste em considerar a região inferior da imagem como pertencente ao chão e não contendo obstáculos. Utilizando esta região arbitrária, dados estatísticos podem ser coletados de forma que outras regiões da imagem podem ser classificadas como chão ou obstáculo. Este método é extremamente eficiente e o maior gargalo é a velocidade do meio de transmissão da câmera ao transferir as imagens. Os maiores problemas destas técnicas são os obstáculos que não se situam imediatamente sobre o chão e obstáculos com o mesmo perfil estatístico assumido para o chão.

Após a extração do chão na imagem, mede-se a distância em pontos para cada coluna da imagem, da base da imagem até o primeiro pontos de obstáculo encontrado. O perfil de distâncias em pontos pode ser convertido em distância métrica utilizando os



dados obtidos da câmera durante o processo de calibração [63].

O perfil de distâncias pode ser então projetado sobre um mapa, dada a posição atual do sistema de medição, acrescentando informações que podem ser utilizadas pelo sistema de navegação para planejar caminhos e desviar de obstáculos. A Figura 14 mostra um exemplo desta abordagem, incluindo um cone de medições a ser inserido no mapa (a câmera se localiza no vértice do cone, com regiões brancas livre de obstáculo e cinza regiões desconhecidas).



Figura 14: Algoritmo monocular na medida de perfis de distância

Normalmente, algoritmos de visão monocular para medição de distância incluem algum procedimento de remoção de ruídos para tornar o chão tão suave quanto possível sem remover o contraste com os obstáculos. Isto significa que imagens de baixa qualidade ou fora de foco (devido a movimentos bruscos ou problema focal da câmera) ainda podem ser utilizados com resultados razoáveis.

A idéia principal de visão monocular para medição de distâncias pode ser simples (extrair chão sobre o qual se pode navegar), mas os algoritmos podem ser aprimorados para lidar com uma série de dificuldades, tornando o uso desta técnica possível mesmo quando a cobertura do chão muda [51] ou ainda em ambiente externos com desníveis e recobertos de folhas de árvores [50].

Ainda assim, alguns problemas envolvendo visão monocular precisam ser resolvidos ou ignorados. Primeiramente, o fato de assumir que os obstáculos tocam o chão geralmente é válido, porém, quando falha, faz com que o sistema de medição assuma que os obstáculos estão mais distantes do que de fato estão, produzindo colisões em potencial. Uma possível solução, sem o uso de uma câmera extra, é tentar processar os



dados incluídos em imagens anteriores, por exemplo, usando algoritmos de fluxo ótico<sup>2</sup> [42, 37].

Além disto, objetos delgados (como pernas de cadeiras, por exemplo) podem ser ignorados pelo sistema de visão pelo processo de filtragem, sendo difícil balancear o processo de remoção de ruídos. Caso o procedimento remova ruído demais, quando existir pouco contraste entre o chão e o obstáculo, será impossível separar corretamente as regiões. Caso remova pouco ruído, medições espúrias serão geradas devido à má extração do chão.

#### 2.4.2: Visão estereoscópica

O objetivo da visão estereoscópica é produzir um mapa de profundidades utilizando duas imagens adquiridas por câmeras em diferentes posições. Se a posição de um dado objeto em ambas as imagens é conhecido, é possível medir a distâncias das câmeras até tal objeto por triangulação. No entanto, o processo de casar um dado objeto em uma imagem com relação a outra não é um problema de simples solução e em geral é computacionalmente complexo.

Normalmente as câmeras são posicionadas lado a lado, tornando o processo de casamento mais simples. É importante notar que os sensores internos precisam ser mecanicamente alinhados e, em câmeras de baixo custo sem o intuito de ser utilizadas nesta configuração, não basta alinhar a estrutura externa. Após o ajuste mecânico, ainda se aplica calibração por programa computacional para correção de perspectiva corrigindo pequenos desalinhamentos [61].

O mapa de profundidades gerado representa a distância de objetos em cada ponto da imagem, em diferentes níveis de profundidade, de forma que, geralmente, objetos mais claros estão mais próximos (Figura 15). O preto total significa que a distância não pôde ser encontrada. Um exemplo de algoritmo de disparidade (processo de casamento completo entre duas imagens e subsequente montagem do mapa de profundidade) é o Birchfield, que apresenta boa relação entre qualidade e desempenho [26, 38].

---

<sup>2</sup>do original em inglês: *optical flow*

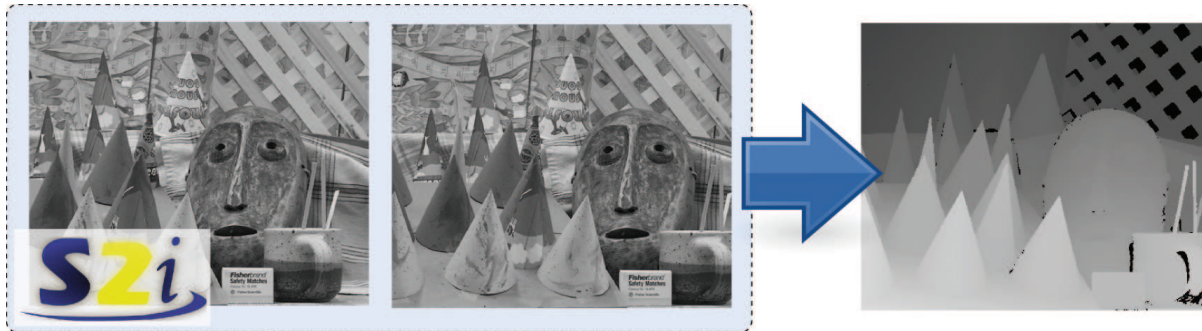


Figura 15: Exemplo de extração de mapa de profundidade

Para criar perfis de distância a partir de mapas de profundidade, é suficiente acumular cada coluna da imagem de profundidade em uma só amostra, tomando os valores máximos de claridade na coluna. Assim como no sistema monocular, os valores dados pelo mapa de profundidade, variando de 0 do mais distante até 255 para o mais próximo, pode ser traduzidos em distância métrica utilizando um simples modelo de projeção, parametrizado pela calibração das lentes. A Figura 16 apresenta os resultados de uma medição utilizando este procedimento.



Figura 16: Algoritmo estéreo para medição de perfis de distância

Já que a abordagem utilizando visão estereoscópica não assume que os objetos estejam sobre o chão, a distância até objetos sem contato com o chão pode ser estimada corretamente, eliminando os problemas desta condição. Além disso, detalhes aguçados (incluindo detalhes aguçados no chão) são bons para o cálculo da disparidade, não sendo necessário o procedimento delicado de balanceamento do filtro de remoção de ruídos. Somente o ruído presente diferentemente entre uma câmera e outra precisa ser removido, tornando possível a medição de detalhes ignorados pelo sistema de visão

monocular (Figura 17).

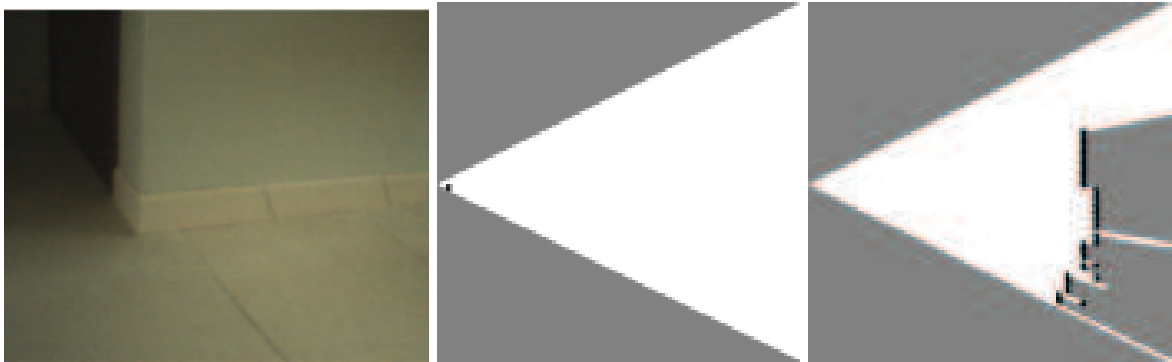


Figura 17: Imagem da câmera esquerda (esq.), perfis de distância monocular (centro) e estéreo (direita)

Por outro lado, o casamento estéreo tende a ser pior em imagens de baixa qualidade ou borradas, já que menos disparidades podem ser encontradas corretamente. Este problema tende a gerar mapas de profundidade com poucas medições, após a remoção de medições de baixa confiabilidade, implicando em obstáculos ignorados. Existem numerosos estudos no sentido de reforçar algoritmos de disparidade para áreas de baixo contraste ou utilizando informação monocular para completar os mapas de profundidade incompletos [68, 56, 25, 67].

A abordagem estereoscópica precisa lidar com muito mais dados e o procedimento de casamento é considerado computacionalmente caro, tornando esta abordagem mais lenta que a monocular. Buscando um melhor desempenho, existem pesquisas de algoritmos mais rápidos [73, 57] ou equipamento de processamento dedicado embutindo os algoritmos [71]. Equipamento dedicado não é flexível e, em geral, financeiramente caro, sendo assim não apropriado quando se deseja testar diferentes técnicas. Por outro lado, mesmo com a melhoria dos algoritmos, é muito improvável que os algoritmos estéreo tornem-se tão velozes quanto os monoculares em um futuro próximo. Desta forma, caso seja necessário desempenho, deve-se abrir mão da flexibilidade e optar por soluções dedicadas.

### 2.4.3: Visão híbrida

Dentro da abordagem híbrida, pouca coisa tem sido pesquisada, e geralmente se trabalha no sentido de completar os mapas de profundidade estéreo com informação monocular [56]. Desta forma, o mapa de profundidade é utilizado para gerar um perfil de distância mais exato.

No entanto, para o caso específico onde somente o perfil de distância é necessário, poderia ser feita a combinação dos dois perfis de profundidade diretamente, uma vez que a combinação de dois vetores unidimensionais é muito mais rápida que operar sobre a imagem bi-dimensional do mapa de profundidade. A Figura 18 exemplifica a junção dos perfis de distância em um único perfil, usando diferentes técnicas.

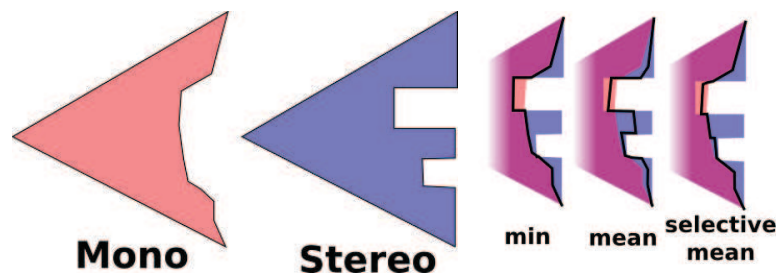


Figura 18: Exemplo de medições para sistemas monocular (esq.) e estéreo (centro) e resultado de três técnicas de junção (dir.)

### 2.4.4: Outras aplicações da visão computacional

Além da medição de distâncias, é possível utilizar a visão computacional para detecção de objetos (incluindo, de forma geral, também aplicações de detecção de pessoas/faces). Em geral, o processo de detecção de objetos passa por três etapas, a segmentação, onde são detectados objetos candidatos e estes são separados do fundo para posterior análise, a extração de características, onde os pontos importantes da imagem são definidos, e a etapa de reconhecimento, onde o objeto, já segmentado, é comparado com a base de conhecimento disponível para detectar qual dentre os objetos conhecidos foi detectado.

## **2.5: Considerações**

As opções de robôs móveis disponíveis no mercado são oferecidas, em geral, para um público de pesquisa, enquanto as soluções que de fato se tornaram comerciais apresentam sistemas bastante simplificados de navegação. Assim, tendo em vista o ambiente de pesquisa, será proposta no capítulo seguinte uma estrutura geral que possa competir em termos de recursos e capacidades com as opções oferecidas pelo mercado.

Uma das capacidades importantes da plataforma proposta é a possibilidade de testes com os algoritmos de visão estereoscópica vigentes, onde possam ser testados ainda algoritmos de navegação e mapeamento. Nas versões mais atuais de robôs móveis do mercado, os sistemas de visão monocular podem casualmente ser tomados como acessório padrão, ou serem disponibilizados via módulos adicionais, enquanto sistemas de visão estereoscópica ainda são bastante raros.



### **3. Proposta de uma plataforma aberta para pesquisa em robótica móvel**

Para tornar o projeto interessante, é necessário que a plataforma robótica satisfaça os critérios econômicos e de flexibilidade, além de ser tecnologicamente atual, realizando testes com sensores modernos, incluindo sistema de visão com suporte à visão estereoscópica.

De fato, o critério da flexibilidade é, neste caso, considerado no sentido amplo, permitindo que absolutamente qualquer aspecto do projeto seja alterável pelos pesquisadores envolvidos no desenvolvimento. Assim, a opção de projetar e construir a plataforma inteira é de certa forma natural, uma vez que se deseja ter um controle fino das opções de implementação do robô.

Por outro lado, em termos econômicos, a construção de uma plataforma robótica é viável se for considerado que o conhecimento necessário para a implementação já existe dentro do grupo de desenvolvimento (formalizado ou não) e que a mão de obra é inserida no trabalho de pesquisa, sendo assim considerada não onerosa.

#### **3.1: Concepção original**

Ao decidir pelo projeto do robô, se ganha liberdade total quanto às funcionalidades a serem implementadas, salvo restrições de recursos. Desta forma, a concepção original da plataforma consiste nas funcionalidades desejadas, incluindo, por exemplo, a forma como será realizada a interface entre o usuário e a máquina.

A Figura 19 demonstra a concepção original da plataforma proposta. Para uma mobilidade total, deseja-se que seja possível utilizar comunicação sem fio, preferencialmente via redes padronizadas que possam se inserir no contexto da Internet. Ao utilizar a internet como meio de transmissão, é possível incrementar o nível de abstração e utilizar protocolos de alto nível, como, por exemplo, o XMPP para troca de mensagens na rede, abstraindo detalhes de protocolos de rede de baixo nível.

Por outro lado, o robô deve contar com um núcleo bem desenvolvido onde possam ser executados todos os procedimentos que dêem suporte às operações de interação com o mundo, sobretudo de locomoção e mapeamento.

Por último, deseja-se pesquisar o uso de visão estereoscópica dentro do contexto de robótica móvel, que é, portanto, enfatizado desde o momento da concepção.

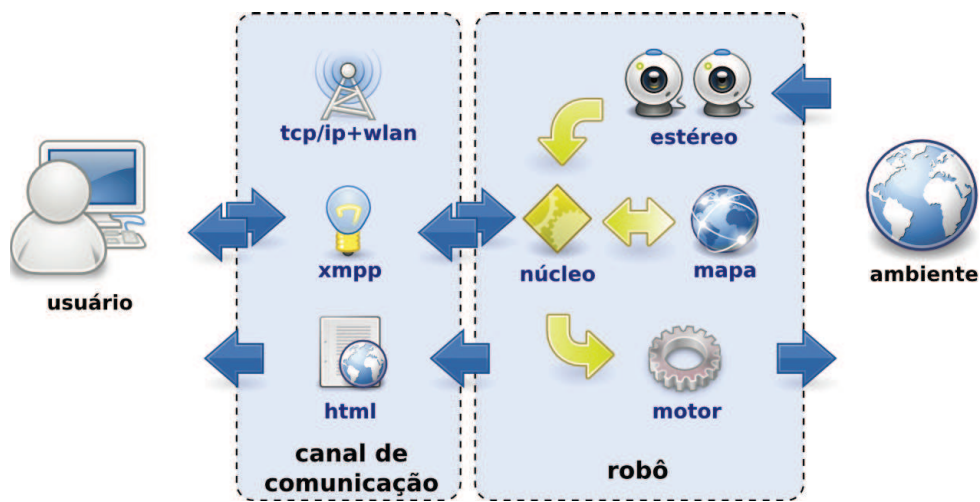


Figura 19: Concepção original da plataforma

### 3.2: Estrutura modular

A concepção da plataforma define o que se deseja que a plataforma seja capaz, não restringindo assim as soluções possíveis para satisfazê-la. A primeira simplificação que é realizada no sentido de projeto de plataforma é assumir que uma arquitetura modular é possível.

Assim, cada módulo é passível de escolhas que, a princípio, não influenciam ou influenciam de uma forma bem determinada os módulos adjacentes na estrutura. A modularização permite a divisão do problema de projeto em subproblemas de menor complexidade, cujas especificações possam ser bem definidas dentro do próprio subsistema, sendo a solução realizada dentro do módulo ou através do bom uso das interfaces. Este tipo de estrutura também permite uma flexibilidade graças à substituição modular (desde que as interfaces entre módulos sejam adequadamente adaptadas).



Para o projeto da plataforma, propõe-se a divisão estrutural entre sistema mecânico, eletro-eletrônico e computacional (Figura 20).

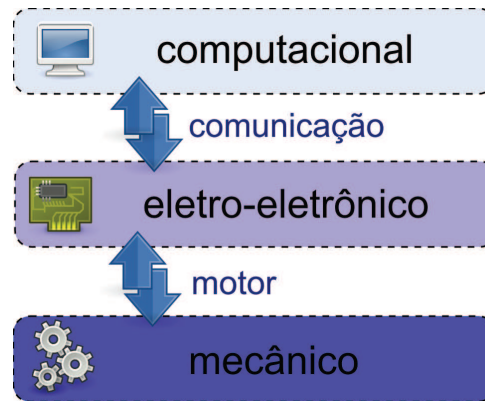


Figura 20: Diagrama estrutural da plataforma contendo os vários subsistemas

O sistema mecânico compreende a estrutura física do robô, espaços de acesso dentro do robô para manutenção e cabeamento, sistema de locomoção e transmissão e é ligado ao sistema eletro-eletrônico a partir de motores (que, de forma sucinta, são responsáveis pela transformação de energia elétrica em mecânica).

No sistema eletro-eletrônico estão inclusas as partes de armazenamento de energia, módulos de potência, assim como a eletrônica necessária para comandar motores. Além disso, contém os sensores de baixa necessidade de processamento. A presença de um microcontrolador neste nível é essencial para transformar os sinais de entrada/saída em dados nos pacotes de comunicação digital, visto que esta é a interface com o sistema computacional.

A partir do enlace de comunicação, inicia-se o processo puramente digital, no sistema chamado de subsistema computacional. Todas as tarefas de alto nível devem ser cumpridas pelo sistema computacional, possuindo este maior poder de processamento e cujos problemas estão na esfera da programação, sob a forma de algoritmos.

Nota-se que a hierarquia explícita no modelo estrutural implica, de certa forma, uma restrição nas operações possíveis de serem realizadas em um dado subsistema. O lado positivo desta compartimentalização é que problemas internos a um dado sistema se tornam invisíveis aos outros subsistemas e, tratando as interfaces dos sistemas com

o devido cuidado, não existe perda das funcionalidades relevantes, ao mesmo tempo em que se facilita o uso daquelas que são de fato relevantes. O lado negativo é que sempre existe uma perda de desempenho e retrabalho/reprocessamento, visto que operações semelhantes acabam sendo executadas em sistemas distintos.

Estas vantagens/desvantagens surgem apenas entre os sistemas eletrônico e computacional, já que eles tratam basicamente da mesma matéria (dados), enquanto entre sistema mecânico e elétrico existem poucas possibilidades de sobreposição.

Dentro de cada módulo, é preciso analisar as opções existentes para a montagem do sistema. A escolha de quais opções usar é, na verdade, um processo que deve ser considerado do ponto de vista global, uma vez que, apesar da concepção modular, muitas vezes existem limitações físicas na combinação dos módulos. Não obstante, cada módulo tem suas opções estudadas separadamente, seguido de breves análises isoladas e por fim, é realizada uma análise integrada das opções sobre as quais o projeto da plataforma foi desenvolvido.

### **3.2.1: Opções para subsistema mecânico e interface**

O subsistema mecânico define como será a estrutura física do robô, de forma que acabará impondo condições de tamanho/peso para todos os outros componentes. A propulsão do robô também está ligada diretamente com o subsistema mecânico, pois é aqui que se define como será realizada a interação entre máquina e ambiente. Ao se tratar de uma plataforma, basicamente se deseja locomoção, deixando outros demais atuadores para interação com objetos fora do estudo, mas lembrando da necessidade de permitir sua posterior instalação como módulos extras.

Assim, várias opções clássicas estão disponíveis para o subsistema mecânico de robôs móveis, sendo que talvez o requisito mais importante é quanto à propulsão do robô. Do ponto de vista de mercado, temos principalmente os robôs com tração diferencial (duas rodas tracionadas e opostas), com ponto de apoio livre ou rodas acopladas tracionadas e robôs bípedes (Figura 21). Aqui se define rodas acopladas tracionadas como o equivalente à tração em quatro rodas, onde se controla a rotação das rodas de um lado de forma única (pois estas estão mecanicamente acopladas).

Tanto se tratando de robôs com rodas quanto de robôs com pernas, existem muitas opções para ambos os casos, existindo trabalhos extensos enumerando várias possibilidades. Alguns exemplos além destes mais comuns são: rodas holomônicas, rodas com tração direcionáveis (a exemplo de automóveis com tração dianteira), robôs hexapods (6 pernas), tetrapods (4 pernas), equilíbrio esférico [59].

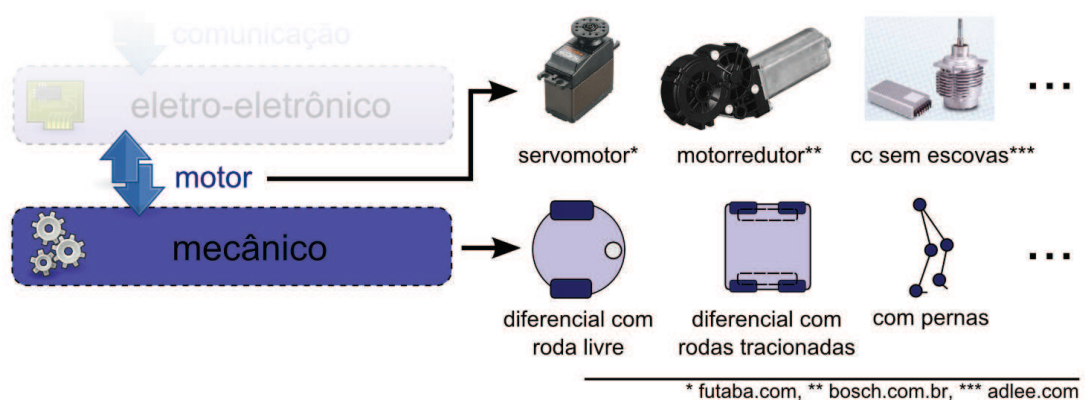


Figura 21: Opções a serem consideradas no projeto do subsistema mecânico e motores

Como o objetivo principal do trabalho não é estudar as formas de locomoção em especial, imagina-se que utilizar as opções tradicionais possa facilitar a implementação, se tratando assim de soluções mais consolidadas. Além disso, sistemas de propulsão com pernas apresentam dificuldades muito maiores associadas, enfatizadas, por exemplo, no número de graus de liberdade (e motores para controlá-los) necessários por cada perna e problemas de estabilidade cuja solução é não trivial, sobretudo do ponto de vista de implementação.

Assim, assume-se que a plataforma deva ser construída tomando como base um sistema de propulsão com rodas, em especial, algum sistema de tração diferencial.

Para a conversão de energia elétrica em mecânica também existem várias opções de motores. Alguns exemplos de motores que poderiam ser utilizados são os servomotores (que já apresentam algum esquema de realimentação interna), motorreductores (motores de corrente contínua com escovas equipados de uma caixa de redução) ou motores CC sem escovas (motores de corrente contínua, sem escovas, equipados de uma caixa de redução).

Os servomotores incluem uma facilidade extra quanto ao uso, pois permitem comando direto de velocidade (ou posição, dependente do servomotor) de forma transparente para o sistema eletrônico, devido a um circuito de controle eletrônico interno. Por outro lado, o controle torna-se fechado e inalterável, perdendo assim um pouco da flexibilidade do sistema. Além disso, a razão entre a potência fornecida e o custo destes motores tende a torná-los uma opção cara na medida em que torques ou velocidades maiores são necessárias.

Motorreduutores são uma opção barata e comum, com o inconveniente de ter um controle de velocidades complexo, principalmente ao assumir que a carga sofrida e o nível de energia do sistema de potência sejam variáveis.

Motores sem escovas ainda não são comuns, mas existe uma forte tendência de aumento da sua aplicação, devido principalmente ao fato de que motores muito leves podem gerar potências elevadas. Uma desvantagem é que esta potência é, em geral, fornecida na forma de rotações elevadas (acima de 20 mil RPM), exigindo sistemas de redução aprimorados e de boa durabilidade. Nota-se que, se a redução correta puder ser utilizada, o torque resultante pode ser tão grande que o torque reativo da carga pode ser desprezado e o controle de velocidade do motor torna-se bastante simplificado. No entanto, não se tem registro de nenhum fabricante brasileiro e até mesmo poucos fornecedores de motores sem escovas no Brasil durante o período de pesquisa (ano de 2008), dificultando assim o acesso a este tipo de equipamento.

### **3.2.2: Opções para subsistema eletro-eletrônico e interface**

O sistema eletro-eletrônico é o responsável pelo armazenamento e fornecimento de energia elétrica para a plataforma, incluindo, se necessário, o circuito de potência para os motores (parte elétrica), além de fazer o processamento primário de todos os sensores e sinais de comando de atuadores (parte eletrônica).

As opções para armazenamento de energia consistem basicamente em diferentes tipos de baterias químicas, como por exemplo: baterias de chumbo-ácido, níquel-cádmio, íon de lítio, polímero de lítio entre outras. As principais diferenças entre as diversas baterias é a relação peso/energia armazenada, custo, simplicidade de operação

(carga/descarga) e durabilidade.

As baterias de chumbo-ácido são a opção mais antiga, cuja utilização já está bastante consolidada, principalmente graças ao setor automotivo. É uma bateria pesada, com boa durabilidade, baixo custo e fácil acessibilidade, devida à escala em que essas baterias são fabricadas.

Baterias de níquel-cádmio apresentam melhor relação peso/energia armazenada, no entanto apresentam inconvenientes com relação ao ciclo de carga e descarga, pois cargas/descargas incompletas tendem a reduzir a capacidade de armazenamento da bateria (chamado efeito de memória). Em geral, não é possível garantir que o ciclo de carga será respeitado durante a operação normal do robô, fazendo com que, na prática, a efetividade deste tipo de baterias seja bem menor.

Tanto as baterias de íon de lítio e polímero de lítio apresentam as melhores relações peso/energia armazenada, no entanto têm elevado custo (incluindo elevado custo do circuito de carga). Este tipo de bateria também elimina o efeito memória. As baterias de íon de lítio apresentavam uma restrição de segurança, pois em situações extremas de operação (temperaturas muito altas, correntes muito altas) elas poderiam pegar fogo e eventualmente explodir. Este risco foi bastante reduzido nas baterias de polímero de lítio, consideradas assim mais seguras. O custo desta opção tende a diminuir, pois estas baterias têm sido cada vez mais necessárias nos ramos da computação e telefonia móvel, no entanto ambas as aplicações acabam produzindo baterias pequenas com capacidades de carga menores que as desejadas, de forma que opções de armazenamento maior (e mais baratas) demoram mais para surgir no mercado.

O circuito de potência é responsável em transformar os sinais eletrônicos de comando, de baixa potência, para comando dos motores, de alta potência, dependendo fortemente do motor escolhido. Assim poderia ser válido incluir o circuito de potência como constando da interface (**Motor**), no entanto, a característica intrinsecamente elétrica do projeto do módulo de potência faz com que pareça mais adequada a inserção neste subsistema do projeto. Em caso de departamentalização de projeto, por exemplo, a decisão do circuito de potência ficaria com o responsável pela parte elétrica do projeto (em acordo com a decisão de motorização).

Para ilustrar alguns circuitos de potência, são apresentados exemplos de circuitos de potência para os motores discutidos na seção anterior: servomotores, motorredutores e CC sem escovas. Servomotores e motorredutores precisam de um circuito próprio (chamados, em inglês, de *driver*), mas que são baratos e possíveis de serem fabricados utilizando componentes eletrônicos simples. Motores CC sem escovas precisam de um circuito de potência mais elaborado, muitas vezes apresentando microcontrolador próprio para a função, uma vez que a ausência de escovas faz com que os terminais do motor tenham que ser alimentados de forma especial.

Por outro lado, as opções da parte eletrônica consistem, basicamente, na escolha de sensores a serem acrescentados (dependendo das especificações de operação) e do microcontrolador a ser utilizado. O microcontrolador centraliza todas as funções do subsistema eletro-eletrônico e torna toda operação com o sistema de controle de alto nível digital. Quanto mais funcionalidades puderem ser embutidas no microcontrolador, mais modularizado o projeto, enquanto a interface entre os sistemas computacional e eletrônico for bem definida.

Algumas opções para microcontroladores podem ser vistas na Figura 22. Os microcontroladores da série Arduino já contém facilidades como circuito oscilador embutido, programação direta pelo cabo de comunicação, programação simplificada utilizando ANSI-C e ferramentas livres. Além disso, em termos de capacidades possuem seis saídas PWM, comunicação I2C de fácil utilização além das várias entradas/saídas digitais e entradas analógicas. Como apresentam um circuito completo, são mais caros que microcontroladores discretos, como por exemplo, o Microchip PIC. Outra característica apresentada pelo Arduino é que, apesar da interface de comunicação com o sistema computacional ser USB, é realizada uma conversão que transforma a comunicação no protocolo RS232, o que apresenta vantagens e desvantagens, que serão discutidas a seguir nesta seção [22].

Os microcontroladores Microchip PIC são uma opção barata, que necessita de poucos elementos para o circuito eletrônico e que, a partir da série 18F4550 apresenta suporte nativo para conexão USB. Apresentam duas saídas PWM, comunicação I2C e várias entradas/saídas digitais e entradas analógicas. A programação, no entanto, pre-

cisa ser realizada por um equipamento especial e não é realizada, portanto, pelo canal de comunicação. Existem muitas opções para compilação dos programas computacionais deste microcontrolador, mas as opções mais consolidadas e com todas as funcionalidades disponíveis não seguem restritamente o padrão ANSI-C e não são abertas (no entanto costumam ser gratuitas) [11]. Outra opção que pode ser utilizada quando é necessário maior poder de processamento para o subsistema eletro-eletrônico são os sistemas baseados em processadores ARM ou FPGA, que podem inclusive integrar também o subsistema computacional (detalhes na próxima seção) [72].

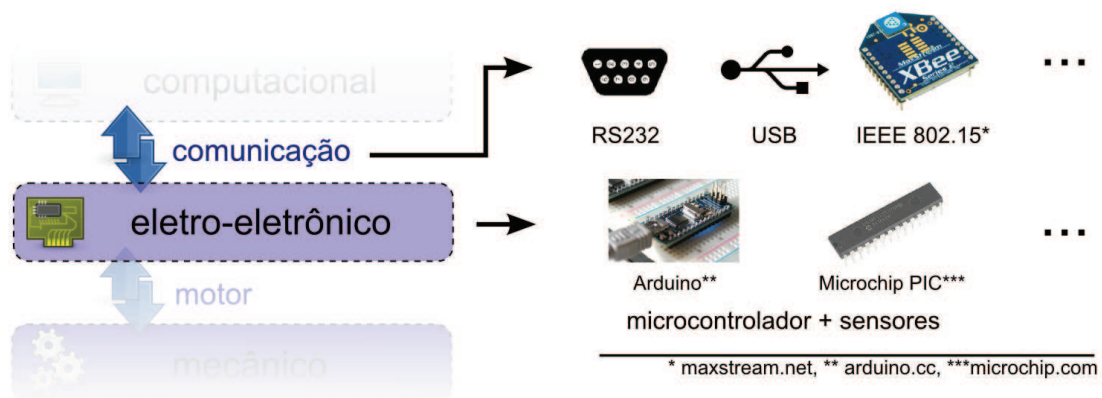


Figura 22: Opções a serem consideradas no projeto do subsistema eletro-eletrônico e comunicação

Os microcontroladores devem se comunicar com o subsistema computacional pela interface de comunicação. Esta comunicação já deve ser toda digital, e as mensagens devem ser bem definidas, restringindo o que pode ser feito pelo sistema computacional, auxiliando assim a modularização dos programas computacionais. Do ponto de vista de equipamentos, a interface depende das capacidades dos microcontroladores do subsistema eletro-eletrônico e do subsistema computacional. Exemplos de interfaces de comunicação são os padrões RS-232, USB e IEEE 802.15 (comercialmente conhecido como Xbee ou ZigBee).

O padrão RS-232 é um conhecido padrão de comunicação serial, que apesar de antigo, ainda é amplamente utilizado, devido à sua simplicidade e baixo custo. No entanto, não prevê um sistema confiável de controle de erro e possui velocidade de co-



comunicação mais baixa que padrões modernos, apesar de em geral, a velocidade ser suficiente para fins práticos. A tendência atual para os computadores IBM/PC é abandonar a comunicação RS-232 em favor de padrões mais modernos como USB, restringindo a comunicação RS-232 via conversores USB/RS232, como no caso do Arduino. Por outro lado, a maioria dos microcontroladores ainda não possui suporte para USB, mas todos apresentam suporte a RS-232.

O padrão USB é também um padrão de comunicação serial, mais sofisticado que o padrão RS-232, apresentando um sistema bastante confiável de detecção de erros e altas velocidades, além de permitir uso de muitos dispositivos em uma arquitetura de barramento, incluindo identificação automática de dispositivos. A dificuldade de uso deste padrão é o esforço de programação exigido para cumprir com os requisitos da comunicação, tanto do lado do dispositivo escravo (microcontrolador, no caso deste projeto) quanto do mestre (sistema computacional). A necessidade de processamento extra dos algoritmos de comunicação (controle de erro, fluxo) é resolvida automaticamente pelos dispositivos que suportam esse tipo de comunicação, de forma que, em termos computacionais, o padrão não pode ser considerado mais pesado que padrões mais simples, como o RS-232, por exemplo.

Existem ainda padrões de comunicação sem fio, como o IEEE 802.15, criado especificamente para comunicação simplificada entre dispositivos sem fio. Como em geral os computadores não suportam o padrão, é necessário o uso de adaptadores em ambas as pontas da comunicação (subsistema eletro-eletrônico e computacional). Num ambiente onde o processamento de alto nível não é embarcado, ou seja, a plataforma móvel não realiza o processamento, sendo esse processamento deixado por conta de um computador fixo, a comunicação sem fio torna-se essencial, sendo então este padrão uma boa opção. No entanto, em arquiteturas embarcadas, onde não existe a necessidade de utilizar comunicação sem fios, esta opção pode tornar o sistema desnecessariamente caro e com menor desempenho que as opções cabeadas.



### 3.2.3: Opções para subsistema computacional

O subsistema computacional é responsável pelo gerenciamento das tarefas do robô e pelo processamento de dados mais complexo da plataforma. Assim, deseja-se poder de processamento aliado a baixo peso e volume. Além disso, podem existir requisitos de protocolos de comunicação suportados e conexões com hardware. É desejável que se utilizem sempre padrões bem aceitos, pois isso facilita o desenvolvimento do sistema devido a maior disponibilidade de ferramentas. Algumas opções são apresentadas na Figura 23.

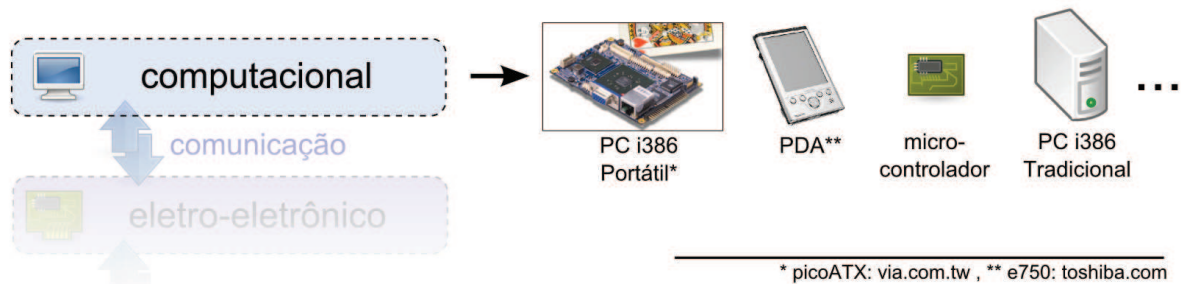


Figura 23: Opções a serem consideradas no projeto do subsistema computacional

Uma opção é a utilização de sistemas baseados na arquitetura IBM/PC. Esta opção é bastante utilizada, pois apresenta uma ótima relação entre custo e poder de processamento, sendo que o tamanho dos componentes vem sendo reduzido cada vez mais e a questão de peso e espaço já não são mais críticas. Um exemplo de sistema compacto utilizando a arquitetura IBM/PC é o formato PC104, que padroniza tamanho e conexões em um sistema ultra-compacto, tipicamente tendo como alvo o mercado industrial e que pode ser utilizado para a robótica [41]. Além disso, mesmo para o mercado consumidor convencional, os sistemas computacionais têm seu tamanho cada vez mais reduzido.

Existem ainda sistemas embarcados com outras arquiteturas, como por exemplo, o sistema baseado no processador ARM apresentado em vários PDAs e placas para sistemas embarcados. Apesar de ser uma boa opção, com baixo consumo de energia e bom poder de processamento, o fato de não ser um produto tão comum quanto PCs

faz com que as opções de infra-estrutura para que se desenvolva os programas para esta arquitetura sejam menos numerosas e o processo de desenvolvimento e depuração torna-se mais lento.

Quando não se faz necessário um processar um volume muito grande de dados, é possível utilizar um sistema com microcontrolador e um programa dedicado, podendo inclusive ser o mesmo microcontrolador utilizado para o subsistema eletro-eletrônico. Esta opção é muito barata, mas apresenta dificuldades, pois o desenvolvimento deve ser extremamente cuidadoso, pois os recursos do sistema são limitados. Outro potencial problema é a pouca disponibilidade de protocolos de comunicação suportados e dificuldade de expansões futuras, no entanto a possibilidade de miniaturização do sistema completo torna esta opção bastante relevante.

Por último, quando muito processamento for realizado, pode ocorrer de não ser possível embarcar o processamento na plataforma. Neste caso, pode-se incluir apenas um sistema mínimo de comunicação (preferencialmente sem cabos e de alta velocidade) e realizar o processamento em uma estação não embarcada, utilizando um PC tradicional de alto desempenho ou computadores ainda maiores. Existem controvérsias sobre se neste caso pode-se considerar o robô verdadeiramente autônomo quando não apresenta processamento computacional embarcado, no entanto, para todos os efeitos, segundo a arquitetura aqui proposta, o sistema como um todo seria considerado um robô autônomo, no qual o subsistema computacional não se desloca com os outros subsistemas, mas mantém a ligação via comunicação.

### **3.3: Análise e testes em protótipos**

Dadas as várias opções descritas anteriormente, torna-se decisão de projeto definir quais serão utilizadas de forma a cumprir com o maior sucesso possível os requisitos da plataforma. Além da análise resumida apresentada, esta escolha se baseia na experiência do grupo de projeto, adquirida na execução de protótipos anteriores e projetos em andamento.

Aqui se resumem os trabalhos realizados no grupo de projeto<sup>3</sup> desde 2005, onde

---

<sup>3</sup>trabalhos realizados no grupo Robota e S2i

puderam ser testadas várias das opções listadas, que culminaram nas análises apresentadas e na escolha das opções para o projeto da plataforma atual. Busca-se ainda o alinhamento com as opções oferecidas pelo mercado, sendo possível a partir delas definir quais decisões seriam factíveis e que funcionalidades são interessantes.

### 3.3.1: Produtos disponíveis

Na Tabela 1 podem ser vistas as escolhas dos subsistemas para as plataformas disponíveis no mercado, conforme disposto no capítulo 2. Nem todas as informações são abertamente disponíveis nos informativos das empresas, sendo necessário consultar minuciosamente os prospectos ou requisitar um detalhamento à equipe de suporte técnico. Ainda assim, as informações referentes ao subsistema eletro-eletrônico são omitidas, caracterizando-se aqui uma estimativa quando notificado na tabela.

Tabela 1: Opções escolhidas nas plataformas comerciais

Robô	Subs. Mecânico	Subs. Elétrico	Subs. Comp.
Pioneer	Motorreductor	Microcontrolador + circuito de Potência (estimados)	IBM/PC compacto
PC-BOT914	Motorreductor	Microcontrolador + circuito de Potência (estimados)	IBM/PC convencional
Pekeell	Motorreductor	Microcontrolador + circuito de Potência	IBM/PC compacto
SRV-1 Blackfin	Motorreductor	Microcontrolador AD-BF537 + circuito de Potência	Microcontrolador AD-BF537

Observa-se uma forte tendência para utilização dos motorredutores como opção padrão para a locomoção pelo subsistema mecânico e a aplicação da arquitetura IBM/PC para o subsistema computacional. Além dos dados resumidos da tabela, é importante fornecer ainda as capacidades em termos de sensores para o subsistema eletro-eletrônico, a partir dos quais é possível observar o processo de inclusão dos sensores de medição de grandezas externas aos robôs, mesmo que ainda como opcionais (Tabela 2).

A utilização de câmeras já é bastante comum, no entanto somente o Pioneer apresenta a opção de visão estéreo. Adicionalmente, estão sendo oferecidas algumas

Tabela 2: Sensores disponíveis nas plataformas comerciais

Robô	Sensores pré-instalados	Sensores opcionais
Pioneer	–	toque, medição de distância, bússola, câmera mono, câmera estéreo
PC-BOT914	medidor de distância, câmera mono	– *
Pekeell	medidor de distância	câmera mono, bússola, acelerômetro
SRV-1 Blackfin	câmera mono	medidor de distância*

\*(fornecida especificação de interface para conexão de sensores extra)

opções para bússola e acelerômetros, no entanto, são opcionais nas plataformas estudadas.

### 3.3.2: Protótipos anteriores

O primeiro robô desenvolvido no grupo de estudo que utilizava a arquitetura proposta foi o robô Ajax, em 2005 (Figura 24-esquerda) [33]. Com uma câmera analógica colorida, era capaz de detectar e medir a posição de objetos com geometria e tamanho fixos e simples (como esferas, cilindros e linhas coloridos), enquanto se movimentava no ambiente. Em um período de transição entre câmeras analógicas e digitais, as analógicas ainda contavam com qualidade superior por um preço menor, com o inconveniente da necessidade de equipamentos extra de digitalização de imagens para o computador (usualmente chamado de **placa de aquisição**). O circuito de potência utilizado foi desenvolvido pelo grupo e foi formalizado somente nos trabalhos para a plataforma atual. As escolhas para os vários subsistemas pode ser observada na Tabela 3.

Tabela 3: Opções escolhidas nos protótipos anteriores

Robô	Dimensões	Subs. Mecânico	Subs. Elétrico	Subs. Comp.
Ajax	30x30x30 (cm)	Motorreductor	Microcontrolador PIC + circuito de Potência	PDA- Processador ARM
Antioch	30x30x30 (cm)	Motor de passos	Microcontrolador PIC	PDA- Processador ARM
RF	15x15x15 (cm)	Motorreductor	Microcontrolador PIC + circuito de Potência	PC convencional/ não embarcado

Seguindo em 2006, foi desenvolvido o robô Antioch (Figura 24-centro), com os

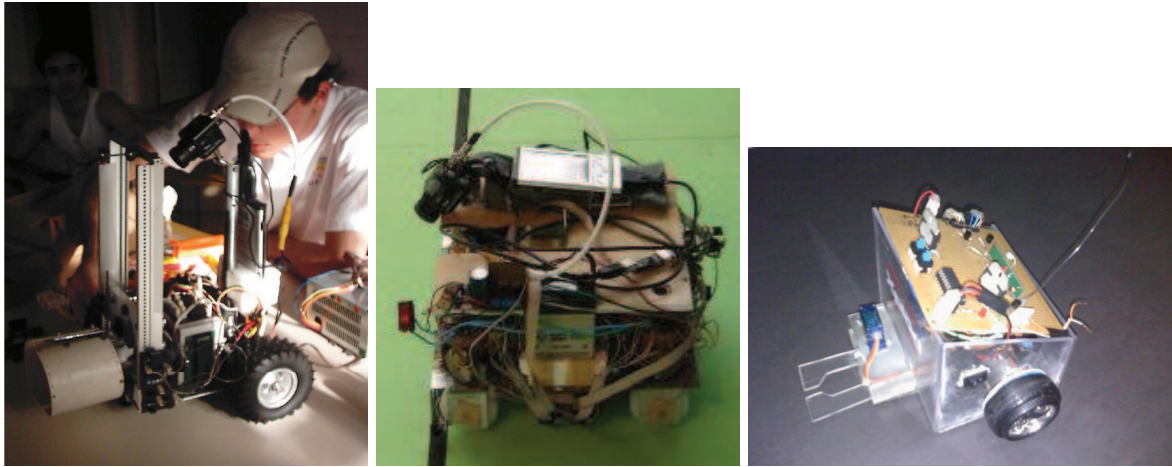


Figura 24: Robôs desenvolvidos que exemplificam combinações das opções de projeto: Ajax-LARC2005 (esq.), Antioch-LARC2006 (centro), RF (dir.)

mesmos princípios básicos do Ajax, com alterações estruturais no subsistema mecânico para que fosse capaz de manipular blocos e empilhá-los, usando também um sistema com quatro rodas para maior sustentação e estabilidade. O sistema computacional sofreu uma total reprogramação na tentativa de tornar o sistema mais confiável.

Por não se tratar de uma arquitetura convencional, o subsistema computacional era de difícil depuração e de lento desenvolvimento, o que atrasou consideravelmente o projeto dos robôs, pois as ferramentas de desenvolvimento precisavam ser construídas ou adaptadas para que pudessem ser utilizadas no processador e sistema operacional não convencionais.

Entre 2007 e 2008, foi projetado um pequeno robô comandado por um PC convencional, via um canal de comunicação por radio frequência (Figura 24-direita). Ao excluir o processamento de alto nível, é possível fazer um robô pequeno, leve e barato. Neste caso o subsistema computacional é considerado não embarcado, e fazia uso de um sistema de visão cuja câmera é posicionada de forma a dar uma visão de topo da área de operação do robô. A rotação das rodas era aqui realimentada por *encoders* especialmente desenvolvidos para o robô e a posição do robô era medida constantemente pelo sistema de visão do subsistema computacional.

### 3.3.3: Protótipo escolhido e modificado

Em 2007 foi desenvolvido o robô El Cabron (Figura 25), criado de forma que fosse possível coletar cubos de madeira de 10 cm em um ambiente estruturado, mas que continha um labirinto, rampas íngremes e um trecho no qual 15 cm do robô ficariam submersos. No projeto do subsistema mecânico, a parte estrutural foi montada de forma a manter somente as partes mecânicas robustas submersas. Para maior robustez, passou-se a usar motorreduzidores automotivos e um sistema de transmissão para que todas as rodas fossem tracionadas, mas ainda seguindo a tração diferencial. Foram adicionados ainda atuadores para carga/descarga de cubos.

No projeto do subsistema eletro-eletrônico, os motores mais potentes exigem mais capacidade de armazenamento de energia nas baterias (constituindo de baterias de chumbo-ácido que totalizam 12V, 6Ah), no entanto o projeto do circuito de potência utilizado no Ajax continua suportando os novos motorreduzidores. Com a necessidade de orientação mais cuidadosa no labirinto, foi incluída no projeto uma bússola eletrônica e sensores de distância, que eram ligados a um microcontrolador PIC. Foram ainda adicionados elementos de interface como visor de cristal líquido e botões à placa do microcontrolador, passando ele a exercer também as funções do subsistema computacional.

Neste caso o robô não apresentava nenhum sistema de visão, pois era possível completar as tarefas propostas com os sensores utilizados.

Por fim, o robô concluído apresentava uma base bastante robusta e confiável, sobre a qual poderiam ser realizados os primeiros testes para a nova plataforma. No entanto, houve a necessidade de modificações no subsistema eletro-eletrônico, mudando sua posição para permitir a expansão física necessária para um subsistema computacional mais potente e exigindo a reprogramação para um novo microcontrolador para que as funções do subsistema computacional não fossem realizadas e fosse adicionado o suporte à comunicação (escolhida aqui a comunicação USB).

Após esta etapa de adaptação, foi adicionado como subsistema computacional um PC ultra-móvel (fabricação pela ASUSTeK Inc., modelo Eee PC(TM)-701), que apre-



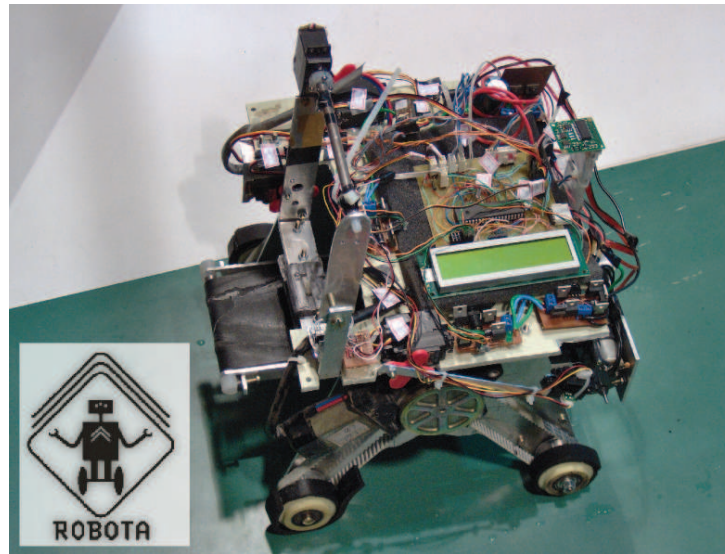


Figura 25: Robô El Cabron projetado para o LARC2007

senta em uma forma extremamente compacta todas as funcionalidades de um PC portátil convencional, com uma certa redução no poder de processamento, mas ainda de uma forma economicamente viável. A possibilidade de fácil desenvolvimento de uma plataforma PC convencional torna esta opção particularmente interessante, incluindo suporte a rede sem fio 802.11, visor LCD colorido e câmera embutida, que são muito úteis durante o desenvolvimento para testes rápidos e modificações diretas no programa do robô. Possui ainda memória não volátil de estado sólido, que permite armazenamento de dados de forma confiável sem perigo de dano devido a choques mecânicos e trepidações, diferentemente de discos rígidos tradicionais (Figura 26).

O subsistema computacional liga-se ao eletrônico via conexão USB e são adicionadas ainda duas câmeras em um arranjo estéreo. Para a modificação do protótipo para adição do sistema de visão optou-se por utilizar duas câmeras CMOS de baixo custo, ligadas via USB ao PC ultra-móvel.

Nota-se que o alinhamento dos sensores das câmeras é importante para melhorar os resultados dos algoritmos de visão estereoscópica, e deve-se verificar se os sensores são alinhados mecanicamente na montagem do arranjo das câmeras. Em câmeras de baixo custo é comum o desalinhamento mecânico do sensor, visto que o propósito das câmeras não inclui atividades que exijam exatidão no alinhamento. Assim executou-se



Figura 26: PC ultra-móvel utilizado

então o alinhamento mecânico dos sensores e estruturalmente as modificações necessárias foram realizadas (Figura 27).



Figura 27: Robô El Cabron após segundo protótipo, modificado especificamente para pesquisa

Assim, as modificações nos programas existentes e testes dos algoritmos foram sendo realizados sobre esta plataforma, que à medida que os resultados foram sendo



produzidos, provou-se por si só um bom projeto inicial de plataforma para testes em visão estereoscópica. Detalhes sobre as implementações e aplicações testadas podem ser vistas no capítulo 4. A tabela 4 descreve as opções utilizadas no robô original e na plataforma modificada.

Tabela 4: Opções escolhidas nos protótipos de teste

Robô	Dimensões	Subs. Mecânico	Subs. Elétrico	Subs. Comp.
El Cabron	30x30x30 (cm)	Motorreductor	Microcontrolador PIC + circuito de Potência	Microcontrolador PIC
El Cabron modificado	30x30x30 (cm)	Motorreductor	Microcontrolador PIC + circuito de Potência	PC ultra-móvel

Considerando o resultado positivo das modificações do antigo robô, tratando esta etapa como um estudo de viabilidade, inicia-se o projeto da nova plataforma, visando atingir com melhor desempenho os requisitos de projeto, incluindo sistema estéreo de melhor qualidade, formalização do circuito de potência, melhor desempenho dinâmico dos motores e espaço para expansões da plataforma em si, assim como instalação de novos módulos.

### 3.3.4: Projeto da plataforma atual

O projeto de uma nova plataforma (realizado no período janeiro-julho/2008), baseada no robô modificado para os testes, inclui as seguintes alterações no projeto original:

- Estrutura modular em chapas de plástico em polipropileno e fusos metálicos;
- Tração diferencial em duas rodas com terceira roda livre;
- Eliminação do sistema de transmissão e utilização de rodas maiores;
- Utilização de apenas uma bateria com capacidade de armazenamento equivalente ao projeto anterior e eliminação do circuito de carga;
- Formalização do projeto da placa de potência;
- Substituição do microcontrolador PIC pela placa microcontrolada Arduino;

- Utilização de câmeras digitais de alta qualidade ultra compactas com lentes de baixa distorção;
- Inclusão de um sensor de aceleração (acelerômetro de dois eixos).

Após os ciclos de modificações, implementação e testes, o modelo final gerado em um programa CAD pode ser visto na Figura 28, assim como a plataforma robótica montada. A tabela 5 resume as diversas opções escolhidas para os subsistemas.

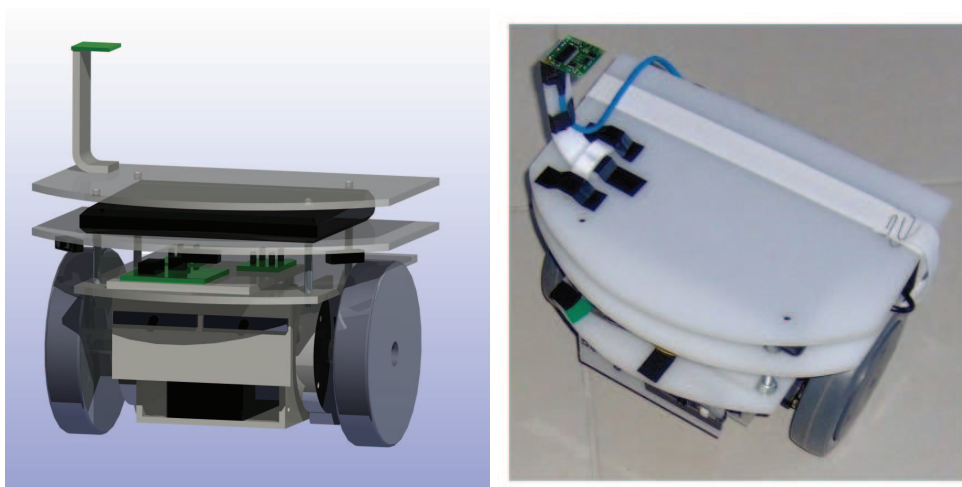


Figura 28: Imagem do modelo CAD e plataforma robótica montada e pronta para funcionamento

Tabela 5: Opções escolhidas no projeto da plataforma final

Robô	Dimensões	Subs. Mecânico	Subs. Elétrico	Subs. Comp.
Plataforma Final	30x30x30 (cm)	Motorreductor	Microcontrolador Arduino+ circuito de Potência	PC ultra-móvel

A estrutura modular permite as futuras expansões desejadas, por exemplo, no sentido de ampliar o espaço para o subsistema eletro-eletrônico, adicionando novos sensores, ou ainda adicionar novos módulos com atuadores (como um manipulador robótico), armazenamento de energia entre outros. Assim, é possível aumentar o tamanho dos andares ou adicionar outros andares trabalhando com os fusos e as chapas.

A tração diferencial em duas rodas e a eliminação do sistema de transmissão visam uma simplificação no sistema mecânico, para redução do número de componentes

e consequentemente, custo e trabalho de montagem. As rodas maiores são necessárias para eliminação do sistema de transmissão e provém uma maior velocidade para a plataforma, possibilitando ainda vários estudos no controle da movimentação da plataforma em velocidades maiores.

Seguindo a redução do número de componentes, reduz-se o armazenamento em uma única bateria de maior capacidade (totalizando 12V/7Ah) e transferência do circuito de carga de energia para o carregador externo ao robô.

O projeto da placa de potência utilizado nos robôs anteriores foi formalizado, com esquemáticos e desenho da placa de circuito impresso com componentes montados.

A substituição do microcontrolador foi realizada para permitir a rápida modificação de seus programas, uma vez que a placa Arduino pode ser programada pela mesma conexão utilizada para comunicação, enquanto os microcontroladores PIC exigem um dispositivo extra para gravação dos programas. Isto permite uma maior flexibilidade do sistema, possibilitando rápidas modificações. Além desta vantagem, o Arduino apresenta facilidades no controle de velocidades dos motorreduzores via PWM, apresentando seis saídas específicas para este fim, enquanto o microcontrolador PIC possuía apenas uma saída<sup>4</sup>, exigindo que fosse realizada toda a programação para o comando PWM (uma vez que são necessárias no mínimo duas saídas PWM para comandar os dois motores), ocupando grande parte do tempo de processamento do microcontrolador nesta função.

Uma desvantagem direta da utilização do Arduino é a utilização da comunicação USB convertida em RS-232, que, apesar de utilizar um padrão moderno de conexão (USB), está limitado às características do padrão RS-232, necessitando assim que sistemas de checagem de erros de transmissão e perda de mensagem sejam implementados.

As câmeras digitais ultra compactas (Figura 29) e o acelerômetro são adições modernas no projeto, sendo que ambas as tecnologias chegaram a preços e qualidade compatíveis com os requisitos da plataforma.

---

<sup>4</sup>existem outros modelos que possuem mais saídas, no entanto estes modelos não possibilitam comunicação via USB

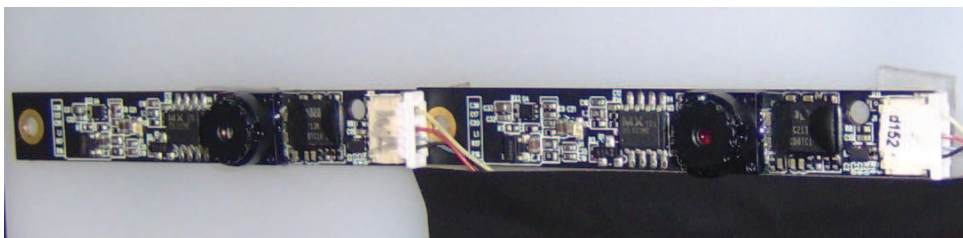


Figura 29: Arranjo de câmeras stereo

A partir da plataforma fisicamente construída seguindo os preceitos acima, concluiu-se a implementação dos diversos subsistemas, que é tratada em maiores detalhes no capítulo 4. Com exceção da programação do microcontrolador, todo o desenvolvimento dos programas computacionais realizados para o protótipo modificado anteriormente pôde ser utilizada, e novas modificações foram realizadas para inclusão de novas aplicações possíveis no novo projeto devido à adição dos novos sensores e novas características de mobilidade (maior velocidade e controle de velocidade simplificado).

### 3.3.5: Protótipos posteriores

Em paralelo ao projeto da plataforma, no período entre janeiro e novembro de 2008, foi desenvolvido o robô móvel WiDe (Figura 30), baseado na mesma concepção da plataforma projetada, mas mantendo a estrutura e opções do sistema El Cabron modificado com sistema de visão com uma câmera apenas (visão monocular).

Adicionalmente, um robô escravo secundário se dispõe sobre a plataforma, contendo um compacto sistema de movimentação, atuador, câmera e microcontrolador, ligados ao mesmo subsistema computacional da plataforma via conexão USB (Figura 31), de forma que o subsistema computacional se encontra não embarcado, realizando a comunicação até o robô escravo por um cabo umbilical. Apesar de poder ser considerado independente, o robô secundário era comandado pela base, que pode ser considerada o robô mestre, de forma a coordenar os movimentos de ambos os robôs, assim podendo considerar o robô escravo como um atuador inteligente do robô mestre. As opções escolhidas para os robôs podem ser vistas na Tabela 6.



Figura 30: Plataforma robótica móvel desenvolvido para o LARC 2008, montagem parcial (esq.) e final(dir.)

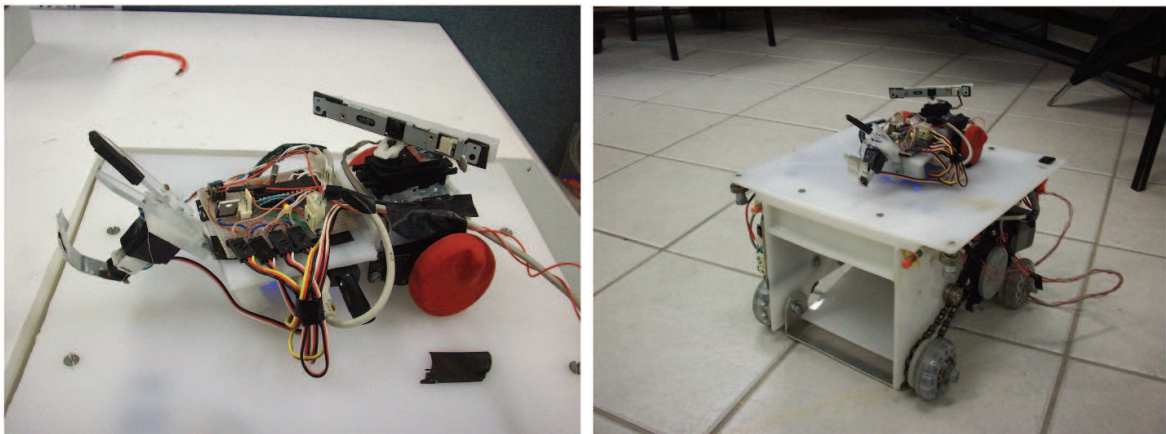


Figura 31: Robô escravo com manipulador e câmera, sobre a plataforma robótica

Tabela 6: Opções escolhidas no projeto WiDe

Robô	Dimensões	Subs. Mecânico	Subs. Elétrico	Subs. Comp.
WiDe Mestre Final	30x30x30 (cm)	Motorreductor	Microcontrolador PIC+ circuito de Potência	PC ultra-móvel
WiDe Escravo Final	15x10x10 (cm)	Servomotor	Microcontrolador PIC	PC ultra-móvel

O robô mestre dispõe de uma câmera fixa e o escravo de uma câmera cujo ângulo com relação à orientação do robô pode variar, de forma que puderam ser testadas estratégias de seguimento e alinhamento de objetos pela câmera e robô independentemente

(os resultados podem ser vistos no capítulo 5).

### **3.4: Considerações**

Dentro das especificações do projeto, foi concebido um conjunto de funcionalidades desejadas, como a comunicação sem fio com o usuário, processamento de visão estereoscópica, capacidade de mapeamento e sistema motor.

Para que tais funcionalidades fossem alcançadas, elaborou-se uma estrutura modular constituída de três subsistemas: subsistemas mecânico, eletro-eletrônico e computacional. O subsistema mecânico se conecta com o eletro-eletrônico através dos motores (primeira interface) e o subsistema eletro-eletrônico se conecta com o computacional pela interface de comunicação (segunda interface).

Para estes cinco itens de projeto (três subsistemas e duas interfaces) existem várias opções, que devem ser escolhidas de forma a satisfazer os requisitos de projeto. Baseado no histórico dos protótipos anteriores, escolheu-se as opções para o projeto atual.

O próximo capítulo descreve com mais detalhes a implementação dos subsistemas, partindo das escolhas definidas neste capítulo.

## 4. Projeto e implementação dos subsistemas

Neste capítulo são descritos os subsistemas da plataforma, detalhando brevemente como foram realizadas as suas implementações no sentido de cobrir as especificações. As opções definidas na etapa de planejamento, como visto no capítulo anterior, trazem implicações diretas na implementação, de forma que, de maneira geral, tenta-se avaliar o resultado das escolhas tomadas com relação às outras opções possíveis, no sentido de validar as estimativas do planejamento.

Os projetos e implementações criados neste trabalho podem ser encontrados no seguinte endereço, incluindo a implementação da versão simulada:

<http://s2i.das.ufsc.br/~erdtmann/stereobot>

E a biblioteca S2i3D, na qual foi baseado o sistema de visão estereoscópica, e portanto também inclusa no endereço acima, pode ser obtida em:

<http://s2i.das.ufsc.br/s2i3dlib/>

### 4.1: Subsistema mecânico

Devido às alterações significativas com relação aos protótipos anteriores já existentes, na tentativa de simplificar o projeto, diminuindo o número de componentes e aumentando a flexibilidade da plataforma quanto à expansões e modificações, o subsistema mecânico precisou de um projeto mais cuidadoso.

Os principais objetivos do projeto deste subsistema é prover uma estrutura modular expansível, ter um sistema de motores com um bom desempenho dinâmico (boa exatidão e velocidade), mantendo o custo baixo e utilizando materiais de boa disponibilidade.



#### 4.1.1: Estrutura modular expansível

A estrutura física que abriga todos os componentes segue um modelo com chapas e fusos, em uma montagem vertical. As chapas definem patamares onde podem ser acrescentados componentes, enquanto que porcas que prendem os fusos às chapas definem a altura entre os patamares.

Esta estrutura permite a compartimentalização dos componentes e pode ser facilmente modificada alterando a posição das porcas e o número de chapas. No entanto, uma dificuldade que limita o crescimento do número de módulos é o aumento da instabilidade devido à elevação do centro de massa, que pode resultar em um pior controle da dinâmica da plataforma (não chegando, no entanto à queda da estrutura).

A Figura 32 mostra duas possíveis configurações da estrutura modular. Para a plataforma projetada, utilizaram-se chapas de plástico PP (polipropileno) e fusos galvanizados, fornecendo uma boa relação entre o custo, peso e durabilidade. O perfil arredondado das chapas na face frontal visa melhorar a mobilidade do robô, evitando que os cantos vivos acabem ficando presos em operações de mudança de orientação.

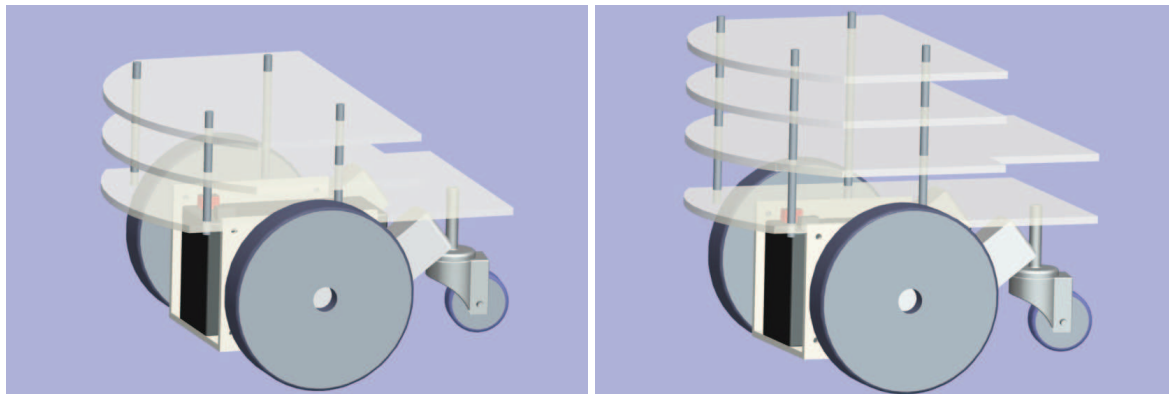


Figura 32: Exemplos de diferentes estruturas possíveis

#### 4.1.2: Motores e reduções

Os motorredutores associam um motor de corrente contínua convencional a uma caixa de redução, de forma que a rotação da engrenagem final seja compatível com as



aplicações tradicionais. Tipicamente, se utiliza um sistema de transmissão para levar o torque rotativo até o sistema de propulsão (eixo + rodas). No projeto da plataforma, optou-se por soldar o eixo das rodas diretamente no eixo do motorreductor, diminuindo os elementos de transmissão ao mesmo tempo em que permite o sistema alcançar maiores velocidades, diminuindo uma etapa de redução na transmissão e utilizando rodas de maior raio.

Por outro lado, os torques exercidos na roda devido à inércia do sistema aumentam a medida que o diâmetro da roda aumenta, a tal ponto que diferenças de cargas e nível de energia da bateria passam a ser fatores importantes no controle (que antes podiam ser ignorados pois o torque reativo sobre as rodas era insignificante). Assim, torna-se mais difícil o controle da cinemática do robô, sendo necessárias técnicas mais aprimoradas para poder alcançar bons resultados. O controle cinemático será brevemente explicado na seção 4.3.5.

## **4.2: Subsistema eletro-eletrônico**

O subsistema eletro-eletrônico apresentou um esforço de integração, centralizando a interface com o subsistema computacional no microcontrolador. Nesta seção serão detalhados o projeto da placa de potência, os vários sensores utilizados e alguns detalhes de implementação, incluindo calibração de sensores e programação do microcontrolador.

### **4.2.1: Armazenamento de energia**

Para o armazenamento de energia, optou-se por utilizar baterias de chumbo-ácido, que fornecem baixo custo e boa durabilidade, apesar de ter um maior peso por unidade de energia se comparado com baterias mais modernas. Por outro lado, ao alojar a bateria na estrutura inferior do subsistema mecânico, o alto peso da bateria ajuda a manter a estabilidade da plataforma, baixando o centro de gravidade da estrutura.

A bateria escolhida é capaz de prover 7Ah@12V, o suficiente para obter um tempo de operação entre cargas de duas horas.

#### 4.2.2: Circuito de acionamento de potência

O circuito de potência converte os sinais de comando do microcontrolador (baixa corrente, tensão) em sinais elétricos para os motores, provendo a potência necessária para que eles operem. Para que os motores possam operar nas rotações direta e reversa, é necessário que o circuito, além de conectar a bateria ao motor com o sinal de comando, alterne os pólos conectados no sentido de inverter o sentido de rotação do motor de corrente contínua. Essa operação é realizada tipicamente com um circuito chamado ponte-H.

A ponte-H opera no sentido de permitir o fluxo de corrente elétrica em um sentido no motor caso um dos sinais de comando esteja ativo, e permitir o fluxo no sentido inverso quando o outro sinal estiver ativo, de forma que recebe, a partir de dois pinos de contato, a informação de qual o sentido de rotação desejado do motor. Caso nenhum sinal esteja ativo, o motor estará livre e não realiza nenhum movimento. O projeto típico da ponte-H não permite que ambos os sinais estejam ativos ao mesmo tempo, situação na qual o circuito seria destruído em um curto-circuito. Assim, é necessário um circuito complementar de proteção para impedir que o circuito final de potência receba o comando dos dois sinais.

O projeto adaptado (Figura 33) substitui o projeto clássico com a vantagem de ser auto-protegido, tornando desnecessário o uso do circuito extra de proteção. Assim, diminuiu-se ao mesmo tempo o número de componentes necessários e a complexidade, mantendo ainda a mesma facilidade de uso. A perda da máxima tensão no circuito final foi de 0.7V, enquanto o circuito tradicional apresenta perda de 1,3V para os componentes operando na mesma faixa de corrente.

Caso ambos os sinais de comando sejam ativos, o motor entra em um estado travado, e apesar de apresentar uma maior energia dissipada que na operação convencional do motor, o circuito mantém-se operante, mostrando a funcionalidade da auto-proteção embutida no circuito.

A Figura 34 apresenta o projeto da placa, em uma visão de CAD e a placa finalizada. Cada placa possui uma ponte-H com capacidade de fornecer potência para um

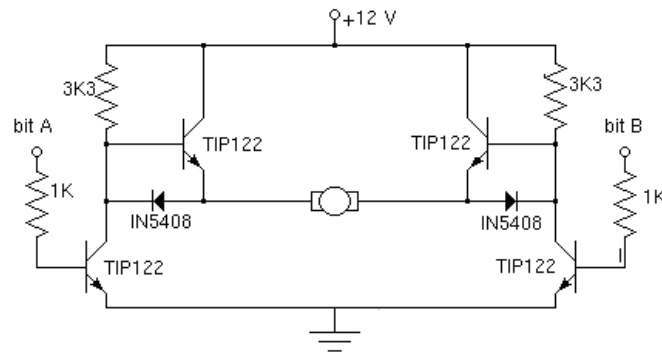


Figura 33: Esquemático da ponte H para acionamento de motor CC

motor operando a até 50V e 5A, mas também foi construída uma placa unificando duas pontes H para o controle de dois motores do mesmo porte.

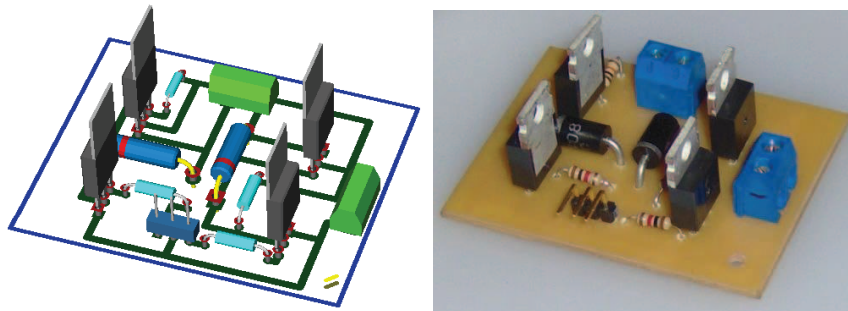


Figura 34: Placa de circuito impresso projetada para o esquemático

Para fornecer velocidade variável, utiliza-se uma técnica chamada modulação da largura de pulso (do inglês *PWM - Pulse Width Modulation*). É possível ampliar a corrente máxima suportada pelo circuito a até 10A substituindo os componentes de potência.

#### 4.2.3: Placa lógica e sensores

A placa lógica inclui o microcontrolador e as conexões para os vários sensores e atuadores. Conforme descrito anteriormente, para o comando dos motores, devem ser reservadas quatro saídas, duas para cada motor, possibilitando assim o controle da direção de rotação do motor. Essas saídas devem ser capazes de gerar sinais PWM,

seja devido a uma capacidade própria do microcontrolador (em inglês, chamado *hardware PWM*), seja pela programação (em inglês, chamado *software PWM*). A opção para o subsistema eletro-eletrônico escolhida é o uso da placa Arduino (Figura 35), que é capaz de definir seis sinais PWM, satisfazendo facilmente esta necessidade. No caso da placa lógica com outros microcontroladores, por exemplo, o Microchip PIC, o sinal PWM precisa ser construído pela programação.

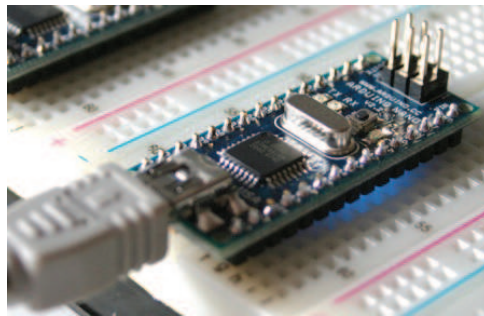


Figura 35: Placa Arduino Nano conectada com cabo USB [22]

Os sensores montados na placa dividem-se, resumidamente, entre sensores digitais e analógicos, sendo os primeiros mais modernos e, em geral, menos susceptíveis a ruídos de medição. Os sensores montados na placa lógica do projeto incluem:

- Três sensores de proximidade - analógicos;
- Acelerômetro de dois eixos - analógico;
- Bússola - digital.

As câmeras, apesar de serem consideradas sensores, devido ao processamento necessário para extrair informações, ficam inclusas no subsistema computacional. A exatidão dos sensores digitais tende a ser maior, no entanto são, em geral, microcontrolados e acabam se tornando mais caros que as versões analógicas. As leituras dos sensores digitais são recebidas através de um barramento de comunicação digital, no caso utilizado o I2C (do inglês: *Inter-Integrated Circuit*), enquanto os dados dos sensores analógicos são coletados nas entradas analógicas e convertidos pela placa lógica em dados digitais.

Independente do fato dos sensores serem analógicos ou digitais é necessário realizar o processo de calibração dos sensores a fim de reduzir o erro de medição, sobretudo removendo tendências de medição e tratando os dados para manter a relação das medições com alguma grandeza física.

A calibração dos sensores de distância segue a curva de calibração fornecida pelo fabricante, cujo formato é parametrizado por medições realizadas contra distâncias conhecidas. O sensor e a curva podem ser vistos na Figura 36. Uma vez que as condições de trabalho do sensor não serão alteradas, o procedimento de levantamento da curva foi realizado manualmente, uma só vez, e o resultado foi conferido para os três sensores utilizados, comprovando a validade da calibração [14].

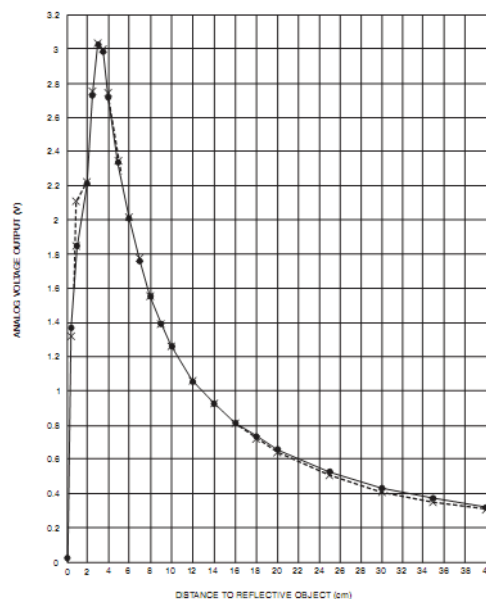


Figura 36: Sensor de distância SHARP GP2D120 e curva de calibração [14]

A curva de calibração indica a perda de sensibilidade (resolução) no sensor para distâncias maiores. De fato, a incerteza de medição para objetos à 80cm é da ordem de 5cm enquanto à 4cm é de 0,1cm, não só causada pela perda de sensibilidade, mas também por uma menor repetitividade. Em geral, este erro não é grave e objetos distantes apresentam também uma menor necessidade de medição de distância exata. A tabela 7 lista os valores lidos nos sensores de distância até distâncias conhecidas, a partir dos quais pode ser ajustada a curva de calibração.

Tabela 7: Valores medidos à distância de referência para ajuste da curva de calibração

<b>Distância referência (cm)</b>	<b>Valor lido (10bits)</b>
100	96
70	152
40	201
20	247
15	298
10	354
7,0	401
5,0	450
4,0	496
2,0	540
0,0	602

A medição de aceleração provida pelo acelerômetro tem ainda um erro muito grande, de forma que o sensor pode ser utilizado para se estimar a ordem de grandeza da aceleração e definir se os movimentos são de aceleração ou desaceleração, sendo mais uma medição qualitativa que quantitativa. A tentativa de integrar o resultado para obtenção da velocidade, por exemplo, leva a erros cumulativos que podem tornar a medida (indireta) de velocidade inviável, como ocorre neste caso específico. Não obstante, o sensor é bastante útil para obter uma realimentação sobre a reação causada pelas ações dos motores.

Além disto, o sensor apresenta o efeito colateral de medir inclinações, uma vez que, ao inclinar o sensor, a componente da aceleração da gravidade nas direções de medição do sensor afeta as leituras do sensor (Figura 37). Assim, realiza-se a calibração da inclinação inicial do sensor, que na verdade é atualizada automaticamente de forma dinâmica para que terrenos inclinados possam ser tratados como planos [6].

Os valores típicos para inclinação inicial podem ser vistos na tabela 8. O microcontrolador, ao comparar constantemente as leituras dos acelerômetros com a atuação realizada nos motores, é capaz de prever situações de colisão, obstrução, inclinação e até mesmo se a plataforma está sendo levantada por uma pessoa. Assim, o sistema de navegação e localização pode decidir sobre quais foram os deslocamentos reais e quais ações tomar no planejamento da trajetória, além de incluir um fator de segurança para

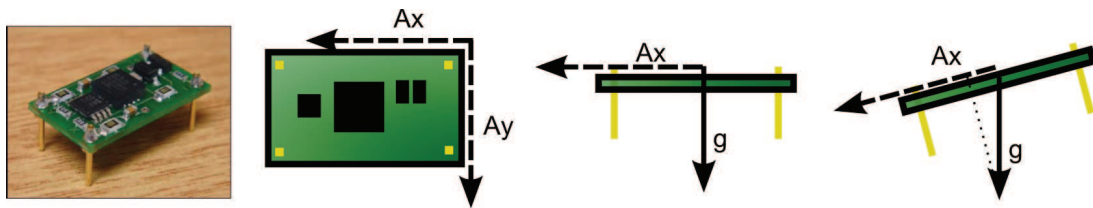


Figura 37: Acelerômetro de 2 eixos e efeito da inclinação na medição (foto à esquerda de [6])

interrupção dos motores em casos de imprevistos.

Tabela 8: Valores iniciais típicos para inclinação do acelerômetro

Eixo	Valor lido (10bits)
X	509
Y	512

As medições da bússola, diferentemente das anteriores, são afetadas por uma série de fatores, incluindo posição geográfica da plataforma, materiais utilizados e estrutura da plataforma (principalmente materiais ferromagnéticos), cabos elétricos (tanto da própria plataforma quanto do ambiente) e outros fatores que influenciem os campos magnéticos naturais da terra. Como não existe forma de garantir que as condições de operação da bússola se manterão constantes, foi necessário criar um procedimento automático de calibração para tornar o processo mais simples e, eventualmente, poder ser realizado sob demanda pelo subsistema computacional [5].

O procedimento automático é executado a partir dos seguintes passos:

1. Comanda-se os motores para atingirem a posição  $320^\circ$  ( $360^\circ$  menos margem de segurança) com relação ao norte;
2. Comanda-se os motores para, com velocidade rotacional constante e translacional nula, rotacionarem lentamente a plataforma em torno de seu próprio eixo, sentido anti-horário;
3. Registram-se todos os valores lidos da bússola, até que se complete uma volta

completa, mais uma margem de segurança;

4. Elimina-se os dados iniciais e finais de forma que se obtenha uma volta completa, de  $0^\circ$  de orientação até  $360^\circ$ ;
5. Assume-se que o ângulo de referência  $0^\circ$  é correto e, como a velocidade de rotação foi constante, os valores de referência são calculados de forma a preencher uniformemente o intervalo entre  $0^\circ$  e  $360^\circ$ .
6. Levanta-se a curva de calibração e gera-se uma tabela de busca (do inglês: *look-up table*) para substituição dos valores lidos contra a referência.

Um exemplo de curva de calibração levantado pelo procedimento descrito pode ser visto na Figura 38. Imaginava-se que as alterações do ambiente seriam significativas a ponto de exigir uma nova calibração ao mudar de ambiente de operação do robô, no entanto, o sensor apresentou-se bastante robusto a essas variações, sendo que novas calibrações tornaram-se necessárias apenas após mudanças estruturais da plataforma.

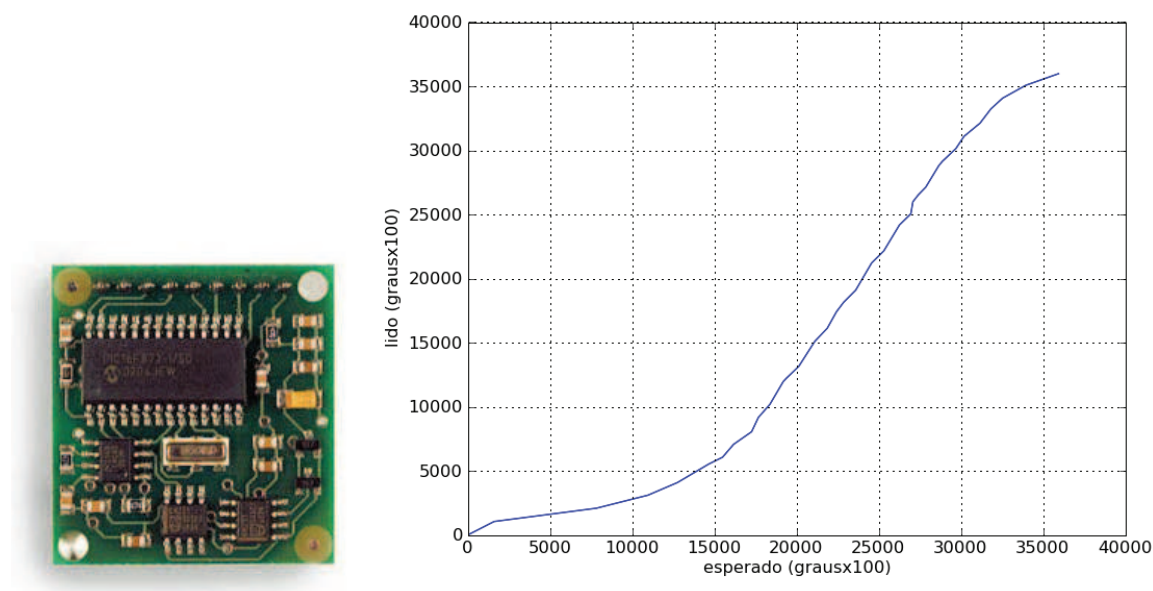


Figura 38: Bússola CMPS03 [5] e curva de calibração obtida a partir do procedimento de calibração automático



Pela curva de calibração, percebe-se uma perda de sensibilidade do sensor na região entre 0 e 130°. O sensor apresenta boa repetitividade, no entanto, devido a esta perda de sensibilidade o erro nesta região chega a 2°, que é um erro bastante aceitável para as aplicações típicas onde existem outros sensores para auxiliar a navegação. É possível realizar uma calibração direta no equipamento da bússola, no entanto esta calibração utiliza apenas quatro pontos e, embora possa eliminar grande parte da perda de sensibilidade, seria ainda insuficiente para eliminar a tendência.

#### 4.2.4: Programação do microcontrolador

O programa geral do microcontrolador consiste em um laço para atualização das leituras dos sensores e troca de mensagens com o subsistema computacional, onde são enviadas as leituras pré-processadas e valores estimados como: velocidade esperada dos motores, se a realimentação global dos acelerômetros detectou alguma situação imprevista (atuação dos motores não realizou o movimento esperado ou houve movimento sem atuação dos motores).

A leitura do sensor mais lento (bússola) tem seu valor atualizado a um período de 20ms, que fez com que se assumisse um período igual ao valor para o laço de operações do microcontrolador.

Ao detectar situações imprevistas, o microcontrolador foi programado para desativar os motores, independente do comando dado pelo subsistema computacional, diminuindo o tempo de resposta para tais situações e aumentando a segurança, uma vez que os comandos de desativação de segurança não poderiam ser ignorados sem a reprogramação do microcontrolador.

Os comandos do subsistema computacional para os motores são recebidos pelo canal de comunicação e, dada a aceleração máxima permitida (como um parâmetro variável), o controlador realiza a transição suave da potência atual para a desejada. Nota-se que, pela ausência de realimentação direta dos motores (como encoders), o comando é realizado em potência, e não em velocidade, sendo que o processamento da realimentação das variáveis medidas externamente ao robô pelos sensores é realizado pelo subsistema computacional, que assim controla como é realizado o controle de

velocidade e posicionamento do robô. Não obstante, a velocidade é estimada no microcontrolador como uma função da potência, e esta estimativa é informada ao subsistema computacional.

Apesar da complexidade extra da utilização apenas de sensores externos, o problema do controle cinemático torna-se ao mesmo tempo interessante e mais poderoso, no sentido de que desta forma a programação do robô não pode assumir que o comportamento comandado terá um efeito conhecido e modelado, mas toda ação precisa ser verificada baseando-se em variáveis externas medidas, comprovando o efeito desejado ou tomando providências para obtê-lo.

### 4.3: Subsistema computacional

Grande parte do esforço do projeto da plataforma incluiu a programação do ambiente para testes de sistemas de visão embarcada, que incluísse também as funcionalidades básicas de navegação e localização da plataforma robótica. De fato, mesmo após a conclusão da programação básica, são realizados os primeiros testes de aplicações com sistemas de visão, procurando, sobretudo examinar a aplicabilidade de sistemas de visão estereoscópica e quais modificações precisam ser implementadas, se necessárias. As implementações dos programas aqui descritos foram realizadas em ANSI-C, com exceção do sistema de comunicação com usuário via XMPP, que foi implementado em Python [13].

As funções do subsistema computacional incluem a obtenção e processamento das informações dos sensores, tarefas de navegação, localização, comando dos atuadores, e interface com o usuário. Todas estas funcionalidades devem ocorrer de forma simultânea e respeitando prazos máximos de resposta, compartilhando os recursos do sistema, ilustrando o cuidado que se deve ter na implementação dos diversos sistemas.

Sobretudo, foi enfatizado o caráter de testes de sistemas de visão embarcados, implementando, baseado nos trabalhos anteriores do grupo, aplicações de visão monocular e estereoscópica, em aplicações de medição de perfis de distância e detecção de objetos sintéticos para localização.

Foi implementado ainda um sistema amigável de interface, com várias possibilidades de fontes de recebimento de comando, passíveis de serem executadas ao mesmo tempo, de forma que o usuário possa ter opções na integração do sistema com outros sistemas, seja localmente na mesma máquina que executa o sistema computacional, seja em qualquer máquina com acesso à rede interna e/ou à Internet, ou ainda para execução de comandos manuais.

#### 4.3.1: Conexão USB

O subsistema computacional deve controlar todas as funções de alto nível, gerenciando a informação obtida dos sensores, e comandando os atuadores via subsistema eletro-eletrônico. A comunicação com o subsistema eletro-eletrônico é realizada pela conexão USB, que foi implementada de duas formas: USB nativo (para placas com microcontroladores PIC ou que suportem USB nativo) e USB emulando conexão RS232.

A implementação utilizada na plataforma final é a conexão USB emulando conexão RS232, em função das capacidades do microcontrolador Arduino escolhido. No entanto, o dispositivo conectado é detectado automaticamente (seguindo as propriedades de auto-configuração de conexões USB), e caso a placa suporte USB nativo, este modo é utilizado para comunicação. Caso nenhuma placa seja encontrada, o sistema entra automaticamente no modo de simulação, que permite o teste dos programas de qualquer computador, sem a necessidade da plataforma. Este modo é extremamente útil para executar testes dos sistemas de visão.

O padrão de comunicação RS232 não define especificamente um sistema de controle de erros, de forma que um sistema simples (checagem de paridade e campo de sequência de pacote) foi implementado de forma que mensagens com erro são descartadas, e a informação do subsistema eletro-eletrônico só será atualizada no próximo ciclo de requisições. O reenvio não é viável para manter com mais confiança o cumprimento de prazos dos diversos fluxos de execução, ainda que, o reenvio de dados antigos torna-se menos útil que o envio de novos dados atualizados.

Nota-se que, no caso de perda de conexão ou perda consecutiva de mensagens, o subsistema eletro-eletrônico é capaz de detectar potenciais problemas operacionais e

desativar o sistema para evitar problemas maiores.

As mensagens recebidas contém os dados contidos na tabela 9. São enviados para o subsistema eletro-eletrônico apenas as mensagens referentes às potências desejadas dos motores.

Tabela 9: Informações recebidas do subsistema eletro-eletrônico

Variável	Descrição
Bússola	última leitura da bússola, em centésimos de grau, valor entre 0 e 36000 ( $^{\circ}/100$ )
Distância esquerda	distância lida pelo sensor de distância posicionado à esquerda (0-100 mm)
Distância central	distância lida pelo sensor de distância posicionado ao centro (0-100 mm)
Distância direita	distância lida pelo sensor de distância posicionado à direita (0-100 mm)
Velocidade estimada	velocidade translacional esperada do robô, baseada no modelo interno (m/s)
Translação Anormal	indica se houve alguma anormalidade durante operação de translação ou com robô parado (booleano)
Rotação Anormal	indica se houve alguma anormalidade durante operação de rotação (booleano)

#### 4.3.2: Programação concorrente

As várias tarefas a serem executadas pelo subsistema computacional foram implementadas de forma a serem executadas paralelamente em múltiplos fluxos de execução, por meio de tarefas periódicas. Isto permite que se tenha uma consistência no período de atualização das diversas variáveis envolvidas e, mais ainda, se pondere o esforço concedido a cada tarefa.

Além disto, a programação concorrente torna possível a exploração dos recursos computacionais modernos, tais como processadores paralelos e com vários núcleos. Apesar deste não ser o caso para os recursos utilizados, a programação realizada está pronta para ser utilizada com vantagens em computadores IBM/PC que possuem tais processadores.

Apesar do comportamento padrão do sistema operacional GNU/Linux utilizado para os testes não fornecer suporte a tempo real, é possível utilizar versões modificadas do núcleo, como por exemplo, as modificações de Ingo Molnár [2, 1], os núcleos Xenomai, RTAI ou RT-Linux, para que se possuam garantias de execução. Todavia, tanto a comunicação quanto à aquisição de imagens das câmeras são executadas sob o protocolo USB que, em última análise, é não determinístico, de forma que continua sendo impossível garantir a execução a tempo.

Ainda assim, considerando que as tarefas essenciais serão exercidas pelo subsistema eletro-eletrônico e que este tem programação cuidadosa no que se refere às características de tempo real, é possível fazer um estudo aproximado dos tempos de resposta (atraso na execução das tarefas com relação ao período estipulado) médios e máximos, que também são comumente chamados de latência da tarefa. No que se refere às tarefas de visão utilizando as câmeras e comunicação, pode-se também contar o número de falhas para cada mil períodos, indicando falha de comunicação USB. No caso de falha, o ciclo da tarefa é ignorado (as variáveis relativas a ela não sofrem alteração) e só volta-se a realizar outra requisição no próximo ciclo.

A opção de suporte a tempo real de Ingo Molnár está no processo de inclusão na árvore de desenvolvimento do núcleo padrão do GNU/Linux, o que torna sua instalação e testes mais fáceis de serem realizados, quando em comparação aos outros núcleos. Devido a esta facilidade de uso, optou-se por realizar os testes em duas versões do núcleo, com e sem o suporte de tempo real de Ingo Molnár, embora não se pretendesse garantir o tempo real, é conhecido que os núcleos de tempo real oferecem menor latência mesmo em aplicações que desconsideram o tempo real, em parte devido ao escalonador de tarefas. Os resultados dos testes podem ser vistos na tabela 10.

Tabela 10: Desempenho das tarefas periódicas (teste executado durante 1 hora de operação)

	Núcleo convencional	Núcleo Ingo Molnár
<b>Latência média</b>	5,2ms	4,9ms
<b>Latência máxima</b>	210ms	13,0ms
<b>Média de falhas de comunicação</b>	2.2/(1000 ciclos)	10.8/(1000 ciclos)

Como esperado, o uso do suporte tempo real diminui consideravelmente a latência, sobretudo a latência máxima. Por outro lado, algumas tarefas, como a aquisição de imagem via USB, não poderiam ser interrompidas, pois neste uma parte da imagem (ou mesmo toda ela) pode ser perdida ou corrompida.

Assim, observa-se que, como o suporte a tempo real tende a interromper quaisquer operações para manter a latência esperada, ocorrem mais falhas de comunicação via USB, resultando em imagens corrompidas. Assim, deve-se analisar qual comportamento é mais desejado no sistema. Como se assumiu aceitável mesmo a latência máxima para o núcleo convencional, optou-se pela utilização deste para que os testes de sistemas de visão pudessem ocorrer de maneira mais confiável.

A tabela 11 ilustra os vários fluxos de execução, descrevendo o período de execução desejado para cada e resumidamente a funcionalidade implementada.

Tabela 11: Fluxos de execução implementados

Fluxo	Período	CPU	Descrição
<b>Bot</b>	50ms	1ms	Base – Comunicação com o subsistema eletro-eletrônico, atualiza os valores lidos dos sensores e o modelo interno de posição da plataforma, envia comandos para os motores e executa o controle de trajetória
<b>Nav</b>	500ms	15ms	Navegação – atualiza o mapa, com base nas leituras acumuladas dos sensores, (re)calcula o caminho até o objetivo
<b>Loc</b>	500ms	10ms	Localização – atualiza a posição atual da plataforma baseado em realimentação externa, de forma a melhorar a estimativa a partir do modelo interno
<b>Cam</b>	100/250ms	45/100ms	Câmera – Realiza as operações do sistema de visão, incluindo medição de perfis de distância, localização de marcos conhecidos. Os valores de período de 100/250 ms são para visão monocular e estereoscópica, respectivamente
<b>Main</b>	sob demanda	N/A	Fluxo principal – inicia os outros fluxos, espera comunicação de programas externos e retorna a resposta das requisições. Finaliza os outros fluxos de forma controlada no caso de requisição de desligamento

Os períodos utilizados são somente valores típicos, que inclusive são ajustados dinamicamente. Por exemplo, em uma operação de alinhamento com um alvo reconhecido pelo sistema de visão, o período do fluxo **Cam** pode ser reduzido. Da mesma forma,

ao alterar os parâmetros do sistema de visão estereoscópica é possível obter tempos de execução maiores ou menores, que são tomados como base para o ajuste de período, que assim é variável quando existe o chaveamento de modos.

A coluna "CPU" da Tabela 11 diz respeito ao poder de processamento necessário dentro do período de execução para executar completamente a tarefa (com o sistema computacional utilizado). Ao fazer a razão CPU/Período e somando todas as linhas, tem-se que a ocupação do processador é próxima de 60% do tempo, indicando que existe disponibilidade de processamento para alguma expansão desejada.

Devido ao número de fluxos de execução, um cuidado especial precisa ser tomado no sentido de evitar o compartilhamento de dados e recursos e, quando necessário, realizar o devido controle do intertravamento dos fluxos e impedir que a interação entre os fluxos gerasse uma parada operacional.

#### **4.3.3: Mapeamento e planejamento de trajetória**

Foi implementado um modelo de mapa vetorial, onde os obstáculos são tratados como segmentos de reta em um mapa bi-dimensional. Assumindo que os trajetos percorridos pelo robô sejam caracterizados por apenas movimentação em um mesmo patamar, não existe perda em planificar o mapa e, mesmo se futuramente for possível a navegação em vários andares em um prédio, por exemplo, é possível seguir uma abordagem multi-camadas bi-dimensionais antes de generalizar o mapa para tridimensional.

Assim, o mapa constitui de vários segmentos de reta encadeados, formando a lista de obstáculos. Posteriormente foi adicionada a lista de marcos (ou localizações), que será melhor detalhada na seção 4.4.4. A posição atual do robô neste mapa é armazenada como as duas coordenadas no mapa (convencionado o nome X para o eixo Norte-Sul e Y para Leste-Oeste) e a orientação como um ângulo de rotação com relação ao eixo vertical, com base no norte (ou X+).

Este mapa pode ser armazenado de forma bastante compacta, a 8 bytes por obstáculo, de forma que um mapa com muitos obstáculos, da ordem de um milhão, ocuparia menos de 10MB de memória, sem requerer nenhum esforço de compactação.

Assumindo uma densidade de 10 obstáculos/m<sup>2</sup> (que dificilmente seria encontrada num ambiente comum), é possível mapear 100 mil metros quadrados de área, considerada mais que o suficiente para as aplicações desejadas. Como os algoritmos de mapeamento são executados em um PC, o mapa todo pode ser carregado em apenas uma fração da memória principal da máquina (que para a plataforma de teste é 512MB).

Para o planejamento de trajetória, o mapa é discretizado em uma dada vizinhança da posição atual do robô. Um valor típico para a distância máxima discretizada é de 30 metros, embora o valor possa ser ajustado dinamicamente e de forma automática pelo sistema. A implementação inicial do processo de discretização envolve percorrer a lista de obstáculos e, caso esteja dentro da vizinhança, o obstáculo é marcado na respectiva célula do mapa discreto. Esta implementação provou-se satisfatória do ponto de vista de desempenho, pois mesmo uma lista grande de obstáculos pôde ser discretizada em muito pouco tempo (<1ms).

O mapa discreto é tratado como um grafo de lugares livres onde cada célula que não foi marcada por nenhum obstáculo é conectada a até quatro células livres (nas direções X+, X-, Y+ e Y-). Ao assumir o peso de cada aresta como unitário, um caminho possível da posição atual até a posição final pode ser encontrado utilizando um algoritmo simples e eficiente de busca em largura de grafos (em inglês: *breadth-first-search*).

Nota-se que, do ponto de vista do grafo, o caminho é ótimo, no entanto, ao gerar a discretização e a vizinhança 4-conectada, as simplificações implicam num caminho não ótimo na prática. No entanto, o resultado é bastante aceitável, pois, em geral, pode-se satisfazer em alcançar os objetivos, mesmo que de forma sub-ótima.

A Figura 39 mostra a interface para navegação, com três exemplos de caminhos para os mapas escolhidos. Os mapas ilustrados foram carregados de um arquivo, no entanto eles podem ser construídos a partir das leituras das distâncias dos obstáculos, como será visto posteriormente. Da mesma forma, um mapa gerado pela navegação em um ambiente pode ser salvo para posterior uso.

Caso novos obstáculos sejam encontrados (ou removidos), as modificações necessárias são executadas tanto na lista de obstáculos quanto no mapa discretizado (evi-



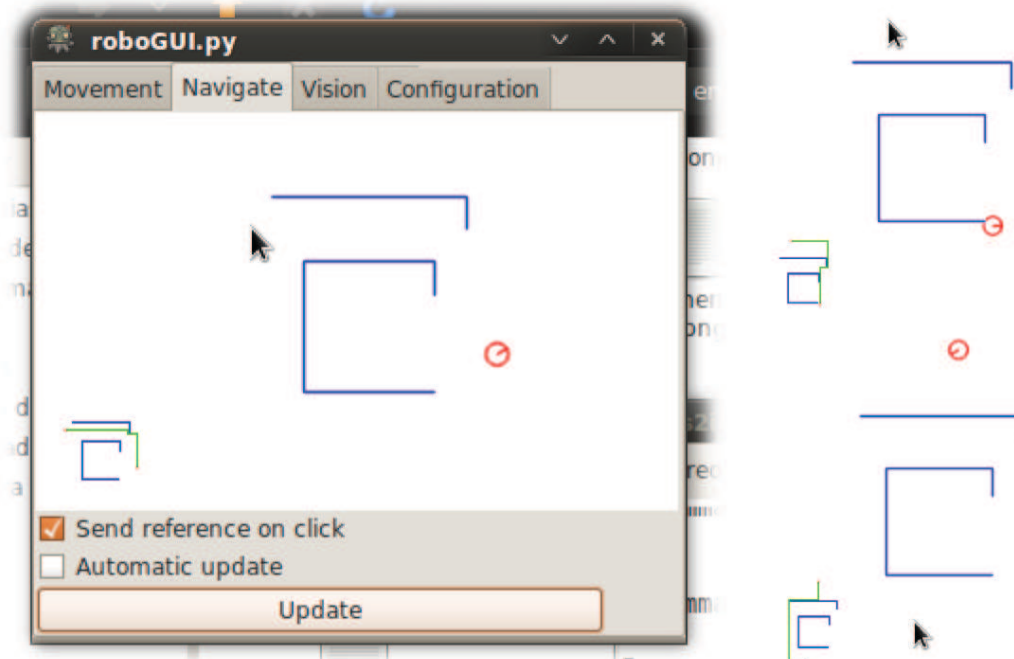


Figura 39: Navegação: mapa vetorial, aceitando entrada de novas coordenadas pelo dispositivo apontador e pequeno mapa discretizado calculando o caminho partindo do robô (círculo) até a posição de destino (cursor)

tando a necessidade de rediscretização).

O tempo máximo de execução para todas as operações sobre o mapa discretizado poderia ser calculado, tendo em vista que, dado o tamanho máximo da lista de obstáculos, tamanho do mapa de discretização, é possível estimar o número máximo de operações executadas. Na prática, no entanto, o tempo de execução é desprezível frente ao lento movimento mecânico do robô. Enquanto a trajetória pode ser recalculada em tempos da ordem de alguns milissegundos no pior caso (rediscretização, e busca em grafos), para movimentar-se um metro o robô leva cerca de um segundo, na velocidade máxima.

#### 4.3.4: Modelagem cinemática e estratégia de controle

O modelo do comportamento da plataforma baseia-se num modelo simplificado, incluindo apenas a cinemática, como detalhado em [58]. A parte dinâmica exerce uma

fundamental importância no comportamento do sistema, de forma que, apesar de não inclusos no modelo, são incorporados na medida que se utilizam sensores para medir grandezas externas à plataforma, compensando os efeitos desconsiderados da dinâmica.

Para o modelo cinemático, o comportamento da plataforma é dada a partir da sua posição  $p_r = (x, y, \theta)$ , definida como a posição do corpo rígido no plano XY e sua rotação no eixo Z (Figura 40(a)). A conversão de velocidades rotacional e translacional nas variáveis da posição é dada em função das velocidades  $u = (u_v, u_\omega)$  (Figura 40(b)). A componente tangencial da velocidade do corpo rígido é definida como a componente da velocidade da plataforma na direção  $\theta$ , obtendo:

$$\dot{p}_r = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} u_v \cdot \cos \theta \\ u_v \cdot \sin \theta \\ u_\omega \end{bmatrix}$$

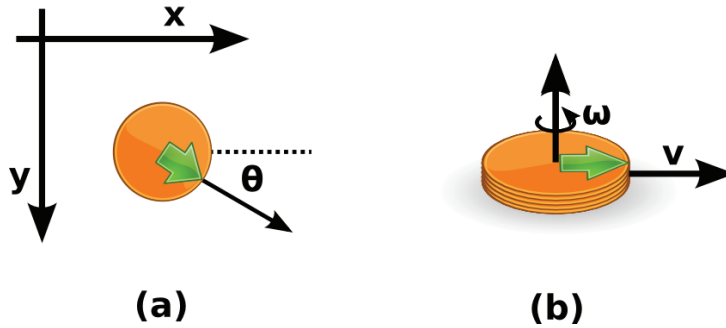


Figura 40: (a) Representação da posição (b) Velocidades do modelo cinemático

Como a formulação acima é diferencial, tem-se as velocidades envolvidas, que passam pelo processo de integração para resultar nas posições:

$$p_r(t_f) = \begin{bmatrix} x(t_f) \\ y(t_f) \\ \theta(t_f) \end{bmatrix} = p_0 + \int_0^{t_f} \begin{bmatrix} u_v(t) \cdot \cos \theta(t) \\ u_v(t) \cdot \sin \theta(t) \\ u_\omega(t) \end{bmatrix} dt$$

Ao utilizar o deslocamento diferencial com as duas rodas, deseja-se representar as velocidades translacional e rotacional como função das velocidades do canto es-

querdo ( $u_l$ ) e direito do robô ( $u_r$ ) e da distância entre rodas ( $l_w$ ), obtendo:

$$u_w = \frac{u_r - u_l}{2\pi l_w}$$

$$u_v = \frac{u_r + u_l}{2}$$

É a partir deste modelo que a plataforma mantém o registro de sua posição no espaço que é consultado, por sua vez, para o planejamento de trajetória. É importante perceber que, ao passar pelo processo de integração, os erros são acumulados, levando a grandes erros. Este problema é agravado quando existem erro crescente na orientação, conforme comentado anteriormente.

Mesmo considerando o modelo dinâmico, a acumulação de erro continua existindo pelo processo de integração, devido às várias não idealidades não cobertas por nenhum modelo. Assim, optou-se por manter o modelo simplificado e incorporar todas os efeitos desconsiderados a partir de medições realizadas por sensores externos ao sistema do robô.

A princípio, é necessário obter ( $u_r, u_l$ ) que, convertidos para ( $u_v, u_w$ ), são utilizados para atualizar o vetor de velocidades da plataforma. Esses valores são assumidos como função direta do sinal de controle dos motores ( $a_r, a_l$ ). O gráfico da Figura 41 ilustra a relação para o motor direito (a mesma relação é usada para o motor esquerdo), notando que a ordenada do gráfico é a rotação do motor que, multiplicado pelo comprimento da circunferência da roda formam a velocidade do ponto extremo  $u_r$ .

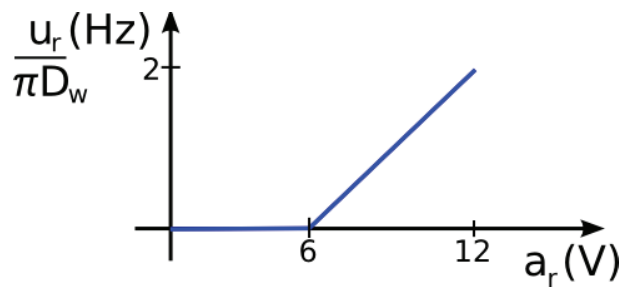


Figura 41: Gráfico (Velocidade dos motores)/(diâmetro de roda  $D_w$ ) contra (Sinal de controle)

Nota-se que esta relação simplificada não leva em conta a dinâmica dos motores, nem derrapagens e colisões que eventualmente ocorrerão. Para corrigir essas irregularidades, compara-se as medições dos acelerômetros e bússola, verificando se a velocidade obtida pelo modelo de motores e pelas leituras dos sensores é a mesma.

Assim, caso o acelerômetro indique que o robô está parado, a velocidade estimada é nula, independente dos motores, assim como movimentos imprevistos (empurrões) atualizam as velocidades de forma equivalente à aceleração instantânea provocada. Seria ainda possível incorporar estratégias de medição de deslizamento via sistema de visão, sendo esse um interessante tópico de estudo para trabalhos futuros.

Devido à inexactidão do sensor utilizado, a medição de aceleração ainda não constitui a solução definitiva para corrigir erros de deslocamento, mas já auxilia bastante na redução de erros.

Erros na orientação, no entanto, são mais críticos, mas eles podem ser limitados pela utilização da bússola para medição direta dessa grandeza, não necessitando passar pelo processo de integração, mantendo assim o erro de estimativa de orientação sempre limitado.

A estratégia de controle de posição passa por um esquema Rotaciona-Avança-Rotaciona, conforme Figura 42. Assim, separa-se em dois controles distintos: controle de ângulo e controle de avanço.

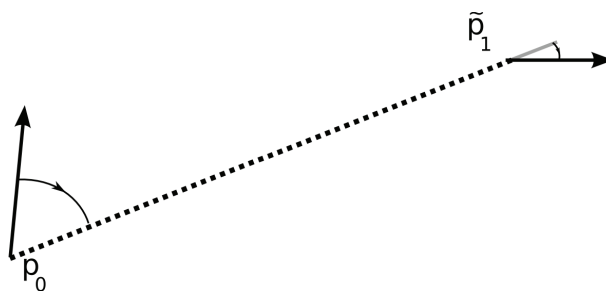


Figura 42: Esquema Rotaciona-Avança-Rotaciona

O controle de ângulo será discutido em detalhes na próxima seção, enquanto o controle de avanço é bastante simples, realizado por um período de aceleração pequeno, progresso à velocidade máxima e desaceleração. Mesmo durante o avanço, a

leitura da bússola é constantemente realizada, de forma que desvios de orientação são devidamente registrados no sistema de integração e, quando excessivamente grandes, corrigidos pelo controle de ângulo.

O perfil de velocidades simplificado correspondente à esse esquema, assumindo posição inicial  $p_0 = (x_0, y_0, \theta_0)$  e final  $\tilde{p}_1 = (0, 0, 0)$  seria aproximadamente dado por:

$$u_\omega(t) = \begin{cases} \pi - \arctan(y_0/x_0) - \theta_0, & 0 \leq t \leq 1 \\ 0, & 1 < t \leq 2 \\ \arctan(y_0/x_0) - \pi, & 2 < t \leq 3 \end{cases}$$

e

$$u_v(t) = \begin{cases} 0, & 0 \leq t \leq 1 \\ \sqrt{x_0^2 + y_0^2}, & 1 < t \leq 2 \\ 0, & 2 < t \leq 3 \end{cases}$$

Conhecendo a relação das velocidades  $(u_r, u_l)$  com  $(u_v, u_\omega)$ , distância entre eixos ( $l_w$ ), diâmetro das rodas ( $D_w$ ) e a resposta dos motores (Figura 41) é possível estimar o desempenho dinâmico da plataforma, que pôde ser posteriormente confirmada como aproximadamente 1m/s de velocidade translacional e 0.5Hz de velocidade rotacional.

#### 4.3.5: Controle de ângulo

Para que a trajetória calculada seja seguida pelo robô é essencial que o controle de ângulo funcione adequadamente, uma vez que erros de orientação pequenos podem gerar grandes erros de posição se não forem corrigidos.

O uso de uma bússola eletrônica permite que se tenha uma medição confiável e externa ao sistema, no entanto, é necessário que exista um sistema de controle que define as ações do robô no sentido de transformar sua orientação atual para uma nova orientação de referência, a ser medida pela bússola.

Assumindo que a velocidade rotacional do robô é definida pela diferença de velocidades entre os motores, pode-se assumir que a orientação esperada do robô é o

resultado da integração da velocidade diferencial. Além disto, pode-se assumir que se conhece a relação entre a velocidade dos motores e a potência que é definida pelo sistema de controle, de forma a montar o modelo completo do sistema com a realimentação do sensor.

Esta abordagem inicial tem alguns problemas, pois desconsidera alguns fatores extremamente relevantes no desempenho real:

- O sistema mecânico apresenta histerese: quando parado, o sistema tende a permanecer parado até que certa energia tenha sido fornecida ao sistema (consequência de folgas, atrito).
- Existe um atraso de medição/comunicação da ordem de 20ms.
- A relação entre potência e velocidade depende dos pequenos obstáculos no chão, da carga sobre a plataforma e do nível de energia da bateria.

Assim, o sistema de controle deve ser robusto a ponto de funcionar nas condições acima. Os resultados para um controlador PID (Proporcional, Integral e Derivativo) simples não foram bons, uma vez que ou o sistema apresentava oscilações muito grandes para ganhos maiores (o sistema integrativo inseria muita energia no sinal de controle) ou não superava os esforços iniciais de movimentação para ganhos menores.

O uso de um controlador chaveado pôde resolver os problemas de histerese e atraso de comunicação. Assim, são utilizados dois controladores PID, o primeiro é ativo quando o erro de orientação é maior que  $60^\circ$  e precisa-se de bastante energia para alcançar rapidamente o objetivo. Ao aproximar-se da região do objetivo, chaveia-se o controlador, assumindo um segundo PID com ganhos menores, fazendo um ajuste fino da posição final. A Figura 43 ilustra duas respostas do sistema.

Na primeira resposta, verifica-se o chaveamento do controlador em cerca de 3,5s, diminuindo o sinal de controle e tendendo a um processo de desaceleração até a parada do sistema, a um erro de aproximadamente  $3^\circ$  da referência, que é considerado aceitável e, sinalizando que o objetivo foi alcançado, o sistema de controle cessa a energia dos motores (erros de posicionamento dentro da faixa de tolerância de  $3^\circ$  são ignorados para

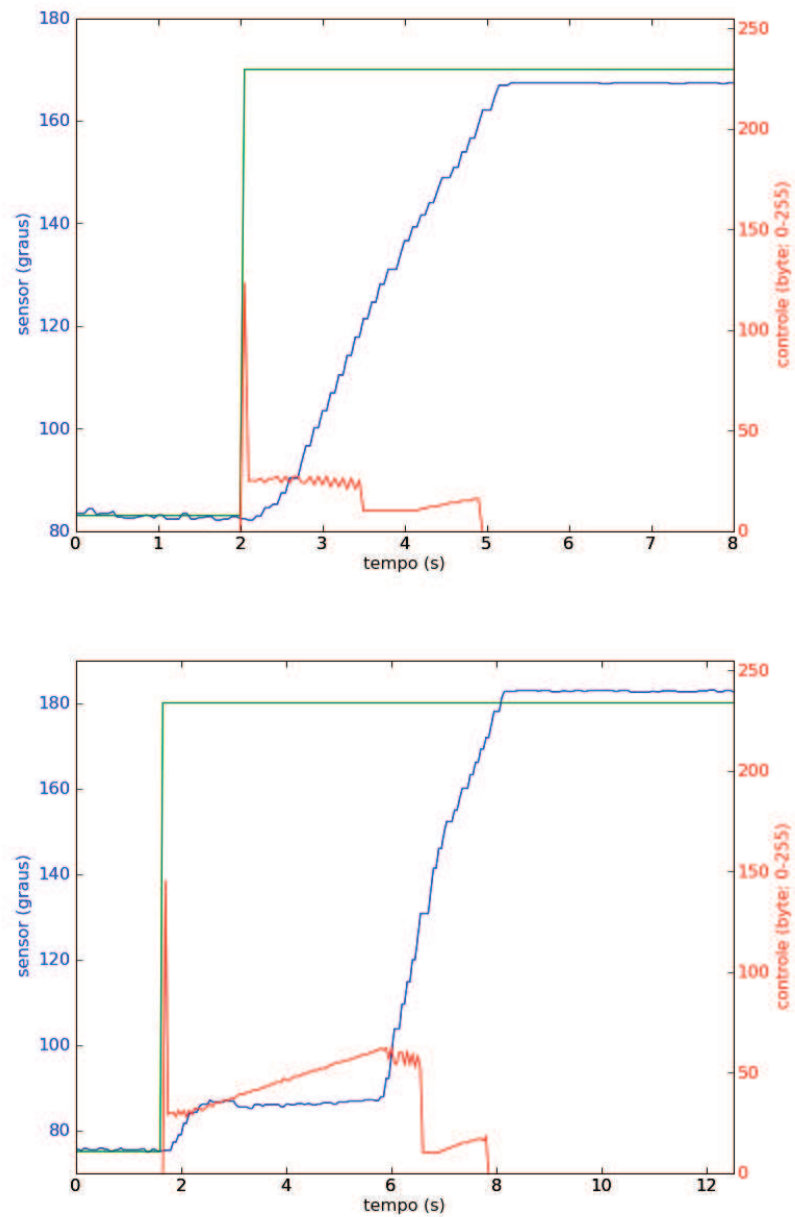


Figura 43: Resultados do controle de orientação para um caso típico e para uma barreira vencida.

permitir movimentação mais rápida evitando ajustes frequentes da orientação). Para a segunda resposta, colocou-se uma barreira impedindo que o robô se orientasse a menos que utilizasse uma potência maior nos motores. Neste caso, observa-se o fator integrativo aumentando a potência de controle até o instante que é possível ultrapassar

a barreira e, logo em seguida, o controle diminui pelo chaveamento. Percebe-se ainda que, devido à inércia do sistema, a diminuição do sinal de controle só acarreta uma diminuição de velocidade quase um segundo mais tarde. Neste caso a orientação final acaba sendo  $3^\circ$  superior a de referência, ainda dentro da tolerância.

Percebe-se que o sinal de controle não está saturado, indicando que a performance do sistema poderia ser ainda melhorada, no entanto, o ajuste dos controladores foi estipulado para que os resultados fossem aceitáveis em uma ampla margem de operação da bateria e rugosidades do chão. A tabela 12 mostra os ajustes para os ganhos proporcional, integral e derivativo para os controladores utilizados.

Tabela 12: Ganhos dos controladores PID - controlador chaveado

<b>Ganho/Controlador</b>	<b>erro<math>&gt;60^\circ</math></b>	<b>erro<math>&lt;60^\circ</math></b>
$K_p$ (proporcional)	0,25	0,03
$K_i$ (integral)	0,085	0,32
$K_d$ (derivativo)	0,05	0,006

O problema da relação potência/velocidade variável é um tanto mais complicado e não pode ser por completo resolvido. Apesar de robusto, o sistema de controle atual não consegue ser efetivo na situação quando se precisa alterar pouco a orientação e o nível de energia da bateria é baixo.

Uma possível solução (que não chegou a ser testada) é estimar a aceleração obtida via acelerômetro e assim avaliar a resposta do sistema mecânico aos sinais de controle, como uma estimativa da carga sofrida, atritos e energia da bateria, resumindo essas variáveis em uma espécie de inércia. Com a estimativa da inércia o controlador poderia se adaptar à situação corrente.

Este tipo de estudo de cinemática de robôs móveis é por si só um tema amplo, o qual somente foi experimentado neste trabalho, com resultados interessantes no uso de realimentação externa e controlador chaveado.



#### 4.3.6: Interação com usuário

Como explicitado anteriormente, um dos fluxos de execução da programação do subsistema computacional tem como objetivo processar requisições externas para o sistema. Essas requisições podem ter várias origens, podendo ser fruto de um programa autônomo de geração de objetivos ou ordens fornecidas pelo usuário humano.

A plataforma deve ser capaz de receber os comandos para controlar sua movimentação e definir pontos de chegada que devem ser processados pelo sistema de navegação. Além disso, é desejável monitorar as variáveis e modificar parâmetros, como por exemplo, o período das tarefas, modo de operação do sistema de visão (entre monocular e estereoscópica), etc.

Assim, toda comunicação foi unificada, de forma que o processo de execução principal pode receber comandos de várias fontes distintas. Como foi realizada a programação para comunicação intra-processos padronizada por encanamentos (do inglês *pipes*), que tem como característica emular leitura/escrita em um arquivo especial para simbolizar a troca de informação entre processos, qualquer interface desejada pode ser obtida, dado o trabalho de programação mínimo necessário. Foram implementadas no decorrer do projeto diferentes interfaces, mostrando a flexibilidade da comunicação interprocessos utilizada, que são listadas a seguir:

1. Interface gráfica de comando: Uma interface gráfica que inclui todas as funcionalidades implementadas, permitindo controle total da plataforma de forma unificada e simples. Originalmente construída para execução apenas local, é possível executá-la remotamente utilizando um tunelamento SSH. (Figura 44)
2. Linha de comando: Em modo texto, funciona como um console de interação. Da mesma forma que a interface gráfica, foi concebida originalmente para uso local, mas pode ser executado remotamente via tunelamento. Tem a vantagem de ser mais veloz quando utilizada na conexão remota.
3. Soquetes TCP/IP: Padrão de comunicação bastante utilizado via rede. Apesar de permitir execução tanto em rede local quanto na Internet, em termos práticos só é

utilizada em redes locais.

4. XMPP (Protocolo expansível de mensagem e presença – em inglês *Extensible Messaging and Presence Protocol*): Protocolo moderno para troca de mensagens via internet, apresenta um formato de mensagens XML e opera no paradigma usuários/lista de contato. Originalmente criado para troca de mensagens entre pessoas, a facilidade de uso está fazendo com que o padrão se torne também útil para troca de mensagens entre máquinas (Figura 45).

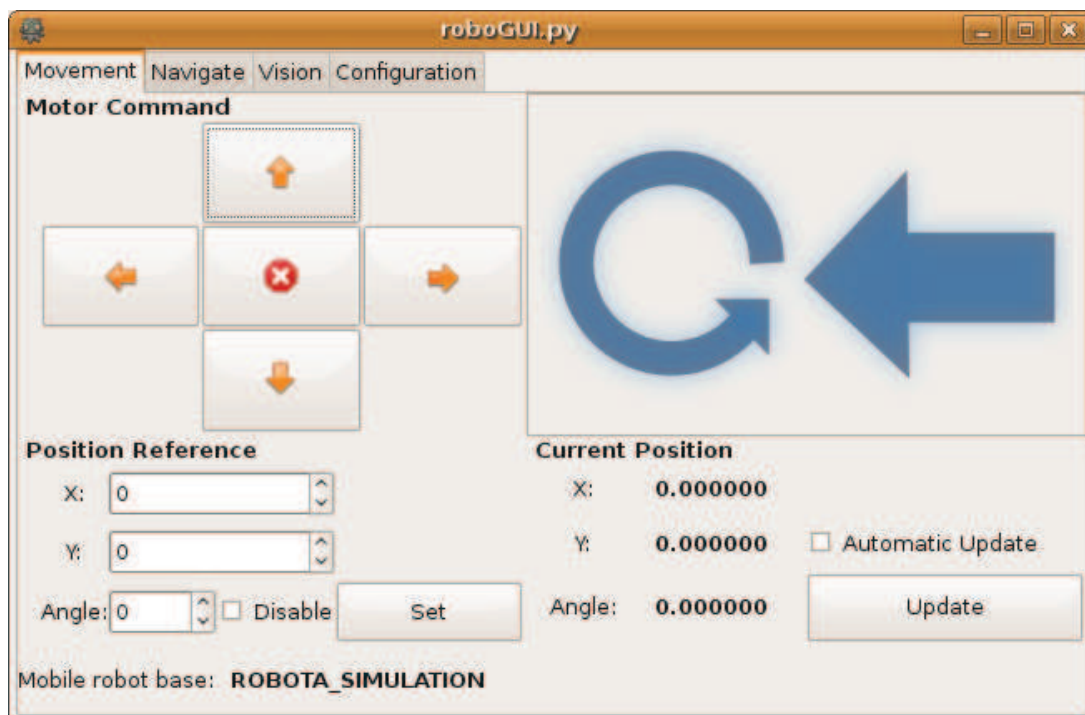


Figura 44: Interface gráfica de comando

Um exemplo notável de uso do protocolo XMPP é no já famoso produto da Google Inc. **GoogleTalk(TM)**, que permite que usuários se comuniquem em um sistema de mensagens instantâneo integrado ao sistema de *e-mails*. A implementação da interface via XMPP permite, por exemplo, que a plataforma apareça na lista de contatos do sistema de mensagens instantâneo e os comandos possam ser enviados diretamente do navegador.

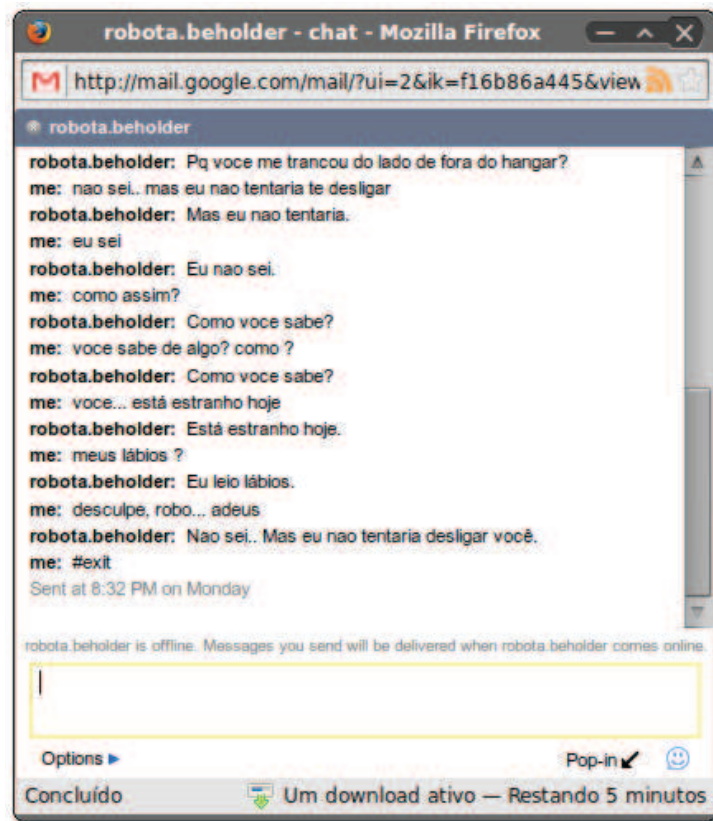


Figura 45: Interface com usuário via XMPP

Como bônus lúdico, integrou-se à interface uma máquina de conversas que permite que, caso a mensagem enviada não seja um comando, o robô responda com uma conversa verossímil. Convencionou-se que comandos são iniciados pelo símbolo cardinal (#), a Figura 45 ilustra o exemplo do uso do comando `#exit` para o encerramento de todos os sistemas do robô remotamente.

#### 4.4: Sistema de Visão

O objetivo primário da plataforma é prover uma base de testes para sistemas de visão embarcados, de forma que a implementação de sistemas de visão é essencial para comprovar as capacidades do projeto. Assim, foram integradas ao subsistema computacional as possibilidades de uso de visão monocular e visão estereoscópica.

Dentre as aplicações possíveis, estudou-se a medição de perfis de distância com

visão monocular, estereoscópica e uma abordagem híbrida que combina o resultado de ambas as medições. Além disso, estudou-se o reconhecimento de símbolos sintéticos e posteriormente foi criado um programa protótipo com a possibilidade de reconhecimento para objetos reais.

O reconhecimento de objetos é aqui utilizado para gerar pontos de referência para o sistema de posicionamento do robô, de forma que os erros gerados pelo sistema de integração de caminho possam ser anulados toda vez que um objeto, cuja posição no mundo é conhecida, for observado.

#### **4.4.1: Visão monocular para medição de perfis de distância**

Ao utilizar visão monocular para medição de perfis de distância, parte-se do pressuposto que os obstáculos permanecem sempre em contato com o solo. Assim, separa-se da cena observada o solo dos outros objetos, que são automaticamente considerados como obstáculos.

Assumindo um terreno plano, a seção a ser observada pelo robô sobre a qual os objetos estão é o solo plano, sendo projetado no sensor da câmera. Nesta projeção que planifica o universo tridimensional, quanto maior a distância vertical na qual o objeto aparece na cena, mais distante do robô espera-se que ele esteja (novamente lembrando que o mesmo encontra-se em contato com o solo).

A parte complicada do processo é diferenciar o que pode ser considerado solo e o que são obstáculos em uma cena qualquer. O primeiro passo é filtrar ruídos que possam atrapalhar na separação, tornando o solo o mais homogêneo possível sem que, no entanto, se elimine obstáculos de tamanho relevante.

A abordagem adotada realiza uma amostragem do solo a partir das últimas imagens obtidas, e a partir dela utiliza um algoritmo de crescimento de semente que marca todas as regiões conectadas à amostra que detenham as mesmas características observadas (em inglês chamado de *floodfill*, ou enchente). Caso a amostra seja reconhecidamente espúria, ela será uma semente pobre e não haverá muitas regiões conectadas a ela, de forma que é possível tomar uma nova amostra na tentativa de melhorar a separa-

ção. Caso nenhuma semente possa ser encontrada em novas amostras, é sinal de que obstáculos devem estar obstruindo a visão, sendo então considerado que os obstáculos estão à distância menor ou igual à mínima do sistema de medição.

Esses passos são ilustrados na Figura 46. Em geral, a região encontrada como solo é marcada de azul para tornar mais fácil a avaliação dos resultados.



Figura 46: Passos de processamento de imagens para visão monocular

Percebe-se que a etapa de remoção de ruído tende a fazer com que obstáculos se mesquem com o solo, como no exemplo da figura acontece com o rodapé da parede. O resultado final disto é que obstáculos semelhantes com o solo podem ser ignorados ou terem medições incorretas de distância.

O processo todo é muito rápido, podendo ser executado com facilidade a taxas de atualização de 25 quadros por segundo (limite das câmeras convencionais) a resoluções de 640 colunas e 480 linhas sem que isso ocupe significativamente o processador e impeça as demais operações do subsistema computacional.

A seguir, pode-se levantar um perfil de distância, medindo uma distância para cada coluna da imagem, lembrando que, quanto mais pontos azuis (que representam o solo) para cada distância, mais distante o obstáculo se encontra. O perfil final medido é encaixado dentro do cone de visão horizontal da câmera, cuja abertura é função do ângulo de abertura horizontal  $\alpha_h$  fornecido pelo conjunto lente-sensor, conforme ilustra a Figura 47.

Até agora, a distância é apresentada como um percentual da altura máxima da imagem, por exemplo, se a contagem de pontos da base até o objeto é de 50 pontos, e a imagem da câmera possui 200 pontos na vertical, considera-se que o obstáculo está a uma distância percentual  $\%D = 50/200 = 0,25 = 25\%$  da imagem. A conversão

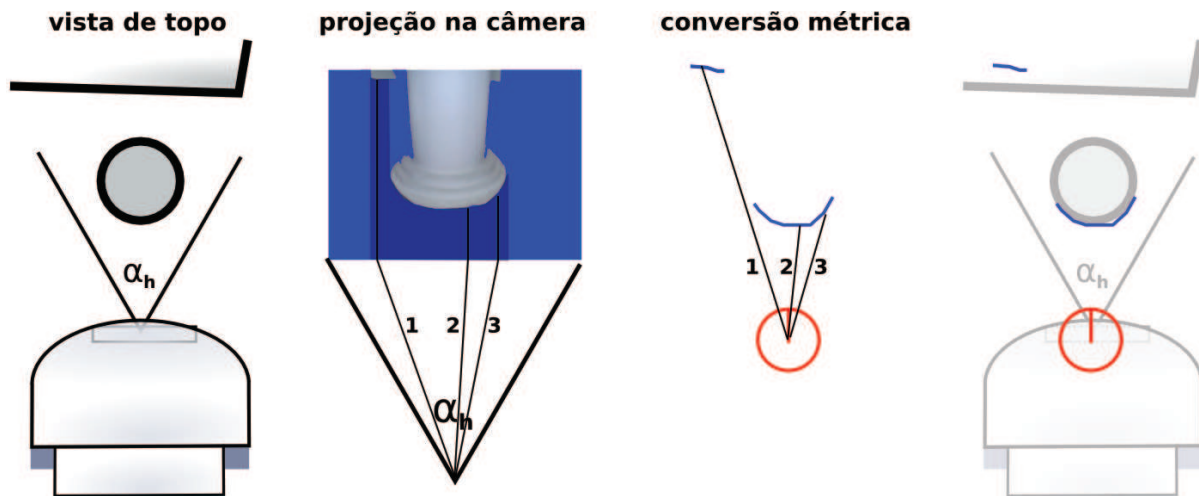


Figura 47: Processo de medição de perfil de distâncias

da distância de pontos da imagem para a distância métrica pode ser feita seguindo a geometria ilustrada na Figura 48.

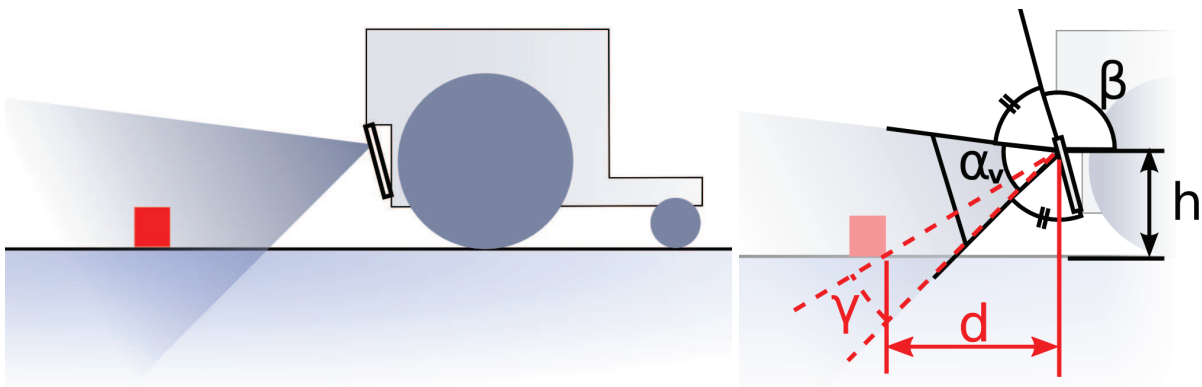


Figura 48: Modelo para medição de distância com sistema monocular

O ângulo  $\gamma$  formado pela projeção do obstáculo na câmera depende das propriedades da lente da câmera, que no caso se resume ao ângulo de abertura vertical, aqui chamado de  $\alpha_v$ , que pode ser facilmente medido e/ou é tabelado. Assim, pode-se concluir pela projeção que  $\gamma = \%D \cdot \alpha_v$ .

Finalizando, a triangulação para a distância métrica é realizada segundo o triângulo retângulo formado entre a câmera e o objeto, cujo ângulo relacionando a distância  $d$  entre o robô e o obstáculo e a altura  $h$  da câmera pode ser encontrado como sendo

$\theta = (\pi - \beta - \alpha_v/2 + \gamma)\text{rad}$  (Figura 49).

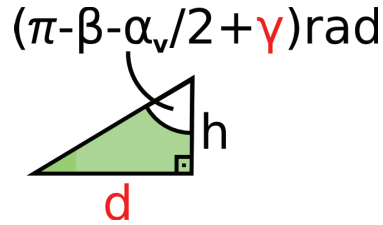


Figura 49: Triangulação final:  $h$ ,  $\alpha_v$  e  $\beta$  são dados,  $\gamma$  medido diretamente e  $d$  obtido pela triangulação

Percebe-se que, à medida que o ângulo  $\theta$  se aproxima de  $\frac{\pi}{2}\text{rad}$ , o triângulo estende-se bastante, sinal da perda de sensibilidade do sistema de distância para distâncias grandes. Dependendo da geometria observada, pode-se inclusive obter valores para  $\theta$  acima de  $\frac{\pi}{2}\text{rad}$ , o que significa que a distância máxima possível (ou linha do horizonte) foi alcançada sem encontrar obstáculos.

#### 4.4.2: Visão estereoscópica

A etapa de eliminação de ruídos na visão monocular, conforme visto, apresenta dificuldades no sentido de eliminar ruídos no solo sem que elimine informações importantes nos obstáculos. Além disso, objetos que não estejam em contato com o solo vão apresentar erros de medidas.

O sistema de visão estereoscópica é capaz de construir um mapa de profundidades da imagem, sendo que mais detalhes, como as heterogeneidades no solo, por exemplo, auxiliam no processo de correlação (ou pareamento) dos pontos, desta forma não sendo tão afetado pelos ruídos. Além disso, a distância medida baseia-se na diferença entre imagens, portanto pode medir indiferentemente distâncias a objetos sem contato com o solo.

Assumindo câmeras alinhadas (ou que já passaram por um processo de alinhamento mecânico e via ajuste de perspectiva com programação), ao estabelecer a correlação de um ponto de uma imagem com a outra, é possível inferir a distância pela diferença das colunas nas quais a correlação foi encontrada (Figura 50).



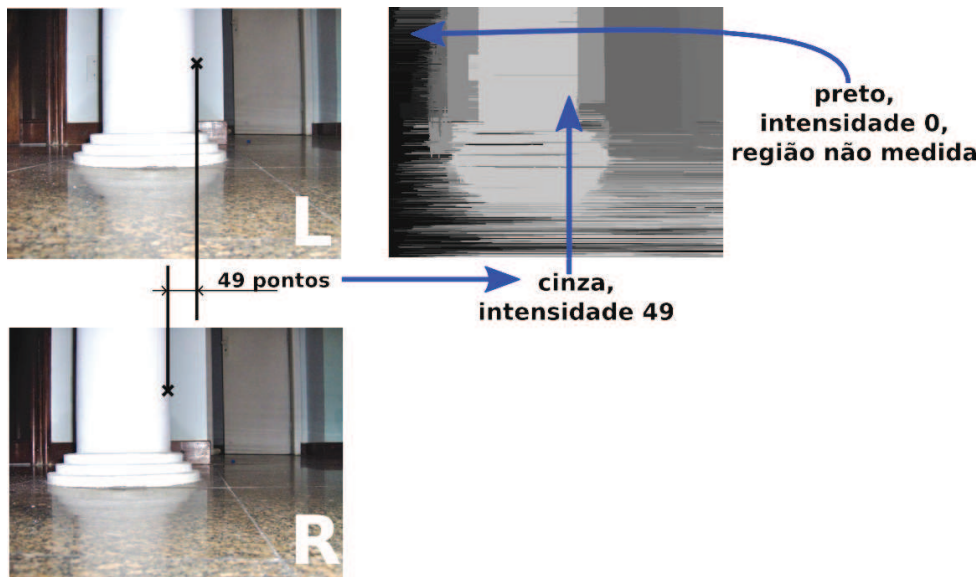


Figura 50: Processamento de imagens estéreo - correlação

Esta distância (em pontos) é então marcada em uma nova matriz, na posição do ponto na imagem de referência (definida arbitrariamente como a imagem esquerda). Caso um determinado ponto não consiga ser correlacionado dentro dos limites desejáveis de erro, assume-se que não se encontrou correlação e nenhuma medida é realizada neste ponto, estipulando-se então que tal ponto será marcado a diferença zero. Esta matriz pode ser visualizada como uma imagem em escalas de cinza, onde zero significaria preto e, a medida que os valores cresçam, a cor tende ao branco.

A parte mais complicada da construção do mapa de profundidades é a correlação entre os pontos. Quando todos os pontos da imagem são correlacionados à outra o processo é chamado de cálculo de correspondência densa. Neste caso, o tempo de processamento é crítico, sobretudo quando se deseja altas taxas de atualização, e cada ponto deve ser comparado com uma vizinhança da imagem seguinte de maneira otimizada. Para o sistema atual, foi possível alcançar taxas de atualização de quatro quadros por segundo, à resolução de 640 colunas por 480 linhas, no entanto o uso do processador é bastante acentuado, tornando esta a atividade principal do sistema computacional [57].

Quando se deseja uma taxa de atualização maior, ou maior confiança na corres-



pondência, pode-se utilizar um algoritmo mais elaborado para somente um ponto (ou alguns poucos pontos), chamado de correspondência discreta. Neste caso, pode-se corrigir desalinhamentos finais procurando correspondências não alinhadas em busca da região com mínima diferença entre as duas imagens. Como somente poucos pontos são tratados, mesmo que o algoritmo seja mais complexo por ponto, em geral o tempo de execução será reduzido e, conseqüentemente, o período de atualização também.

Da mesma forma que na versão monocular, o número de pontos entre o objeto nas duas imagens marca a distância até ele, mas desta vez, quanto menor o número de pontos, mais distante ele se encontra. Novamente pode-se medir a distância inicialmente como um percentual (%D) com relação ao maior número possível de pontos, que é a largura da imagem. Por exemplo, se a distância for de 49 pontos e a imagem tiver 640 pontos de largura, a distância percentual será de  $\%D = 49/640 = 0,07656 = 7,656\%$  [47, 24].

Dado o ângulo de abertura horizontal  $\alpha_h$ , pode-se definir  $\gamma = \%D \cdot \alpha_h$ . Respeitando essas definições, é possível observar as relações ilustradas na Figura 51.

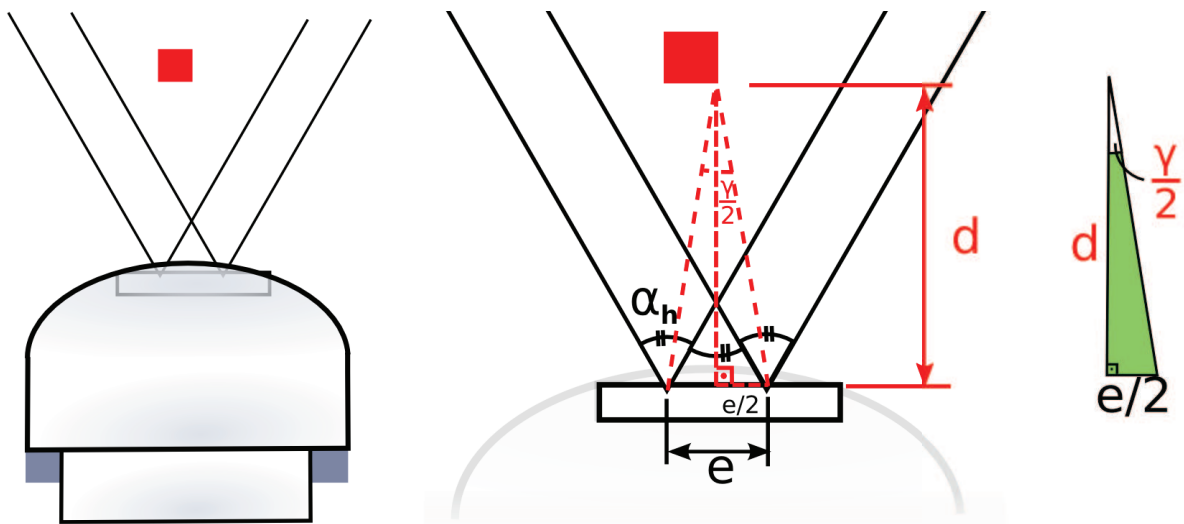


Figura 51: Modelo para medição de distância com sistema estéreo

Desta vez a triangulação é mais simples, leva em consideração apenas a distância entre as câmeras do arranjo estéreo e (implicitamente) o ângulo de abertura horizontal. O processo para montar o perfil de distâncias é bastante parecido com o caso

monocular, mas agora amostra-se uma ou mais regiões de interesse para medir as distâncias, que não mais são consideradas da base até o topo da imagem (Figura 52).

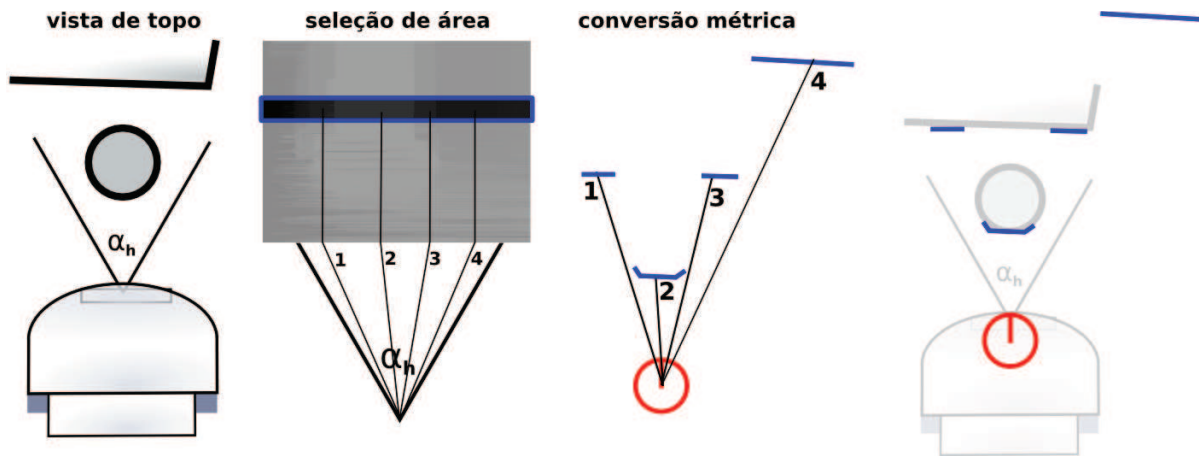


Figura 52: Processo de medição de perfil de distâncias para o caso estéreo

Nota-se que aqui é possível fazer uma melhoria no tempo de execução ao executar o algoritmo de correlação de pontos somente para a área a ser investigada, como um caso intermediário entre a correspondência densa e discreta.

#### 4.4.3: Unificação das medições de distância

É possível obter perfis de distância por visão monocular e estereoscópica, além de existirem as medições dos três sensores de distância. A abordagem que utilizada na implementação é uma abordagem conservadora, que assume que o mínimo de todas as distâncias para mesclar os perfis, no entanto, foram realizados estudos preliminares que apontam que uma solução mais coerente para o problema da unificação de medidas (também chamado fusão de sensores) é a utilização de filtros de Kalman[65].

O filtro de Kalman assume que todos os sistemas de medição do sistema sejam perturbados por ruídos gaussianos unimodais de média zero e que, para cada sistema de medição, se tenha um desvio padrão associado (variável ou não) conhecido. A partir daí o filtro de Kalman produz a melhor estimativa para a grandeza de interesse, mesclando as várias medidas existentes [59].

Alguns testes preliminares do filtro de Kalman fora realizados neste trabalho, em

ambiente de laboratório (não embarcado na plataforma), provendo boas medidas da distância até o objeto de referência. No entanto, para utilização em definitivo do filtro de Kalman seria necessário uma modelagem e estimativa mais corretas do erro de medição associado aos sistemas de medição utilizados, assim a integração do protótipo de testes para a plataforma atual.

Independentemente da estratégia utilizada para mesclar as diversas medidas, por fim obtêm-se um perfil de distâncias medidas, que nada mais são que uma sequência de obstáculos a serem inseridos pelo sistema de navegação no mapa e, assim que devidamente medidos e mesclados, todas as medições vão para a fila de inserção no mapa, e serão incorporados pelo sistema de navegação automaticamente em seu próximo ciclo.

#### **4.4.4: Localização**

O modelo interno da plataforma permite que a posição seja estimada à medida que ela se movimenta. No entanto, mesmo com as realimentações externas pelos acelerômetros e bússola, os erros de estimativa devem ocorrer e, gradativamente serão acumulados.

Os erros mais graves, produzidos por erro na estimativa de ângulo não ocorrerão devido ao uso da bússola, mas erros translacionais ainda existirão, pois não há garantia de que a velocidade estimada pelo modelo interno será de fato a velocidade real. Mesmo que fosse utilizada realimentação da rotação das rodas, ainda existiriam os erros de escorregamento das rodas e pequenas colisões que fazem com que o erro ainda fosse somado. Ao somar os erros durante um tempo suficientemente longo, o erro absoluto pode ser grande demais para ser tolerado.

Assim, técnicas adicionais para localização precisam ser implementadas de forma a limitar o problema do erro de estimativa de posição.

Se objetos cuja posição é conhecida puderem ser detectados pela plataforma, e a posição relativa entre a plataforma e o objeto puder ser medida, é possível atualizar a posição atual da plataforma de forma a limitar o erro, ou, mais precisamente, tornar o erro de estimativa equivalente ao erro da medição da posição relativa. Esta técnica é

conhecida como localização por marcos (em inglês *landmarks*).

Para o sistema de localização proposto, foram consideradas duas etapas. Na primeira, chamada de etapa de aprendizado, as informações sobre os objetos conhecidos são fornecidas e o sistema é levado a aprender a reconhecer o objeto, armazenando as suas características em uma pequena base de dados.

Na segunda etapa, a etapa de operação, a plataforma, operando em condições normais, executa no fluxo de execução de localização uma busca rápida por objetos separando-os na imagem do resto da cena. A seguir, o objeto é comparado com a base de dados para verificar se é um dos objetos conhecidos e qual deles.

Os objetos tratados na implementação inicial realizada são objetos sintéticos, representados por folhas brancas com ícones pretos impressos, exemplificados na Figura 53. Estas folhas podem ser afixadas em locais definidos, como por exemplo, a posição inicial representada pelo ícone **casa** ou o ponto de recarga representado pelo ícone **relâmpago**. Estes símbolos foram utilizados principalmente para facilitar a busca e separação de objetos do resto da cena, e para tanto são constituídos de um quadrado no centro de um círculo, que são ambas as figuras geométricas fáceis de serem reconhecidas por técnicas de processamento de imagem. O tamanho conhecido das figuras também permite que a medição de distância entre o robô e a figura possa ser realizada de maneira confiável.

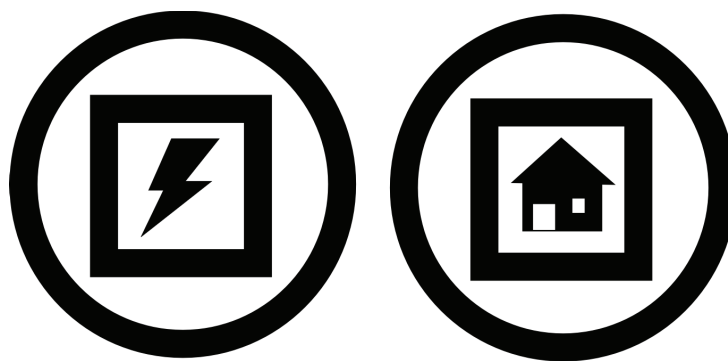


Figura 53: Ícones para localização

Os passos da separação do objeto com a cena passam por uma etapa para encontrar círculos via transformada de Hough circular. Caso algum círculo seja encontrado,

a cena sofre o primeiro corte para apenas a região do círculo, seguido da verificação da presença de um quadrado com a transformada de Hough para retas. Caso este novo critério seja atendido, a cena é novamente cortada para o ícone interno ao quadrado. Os tamanhos do círculo e do quadrado são armazenados para o cálculo da distância, por triangulação.

O sistema de aprendizado/reconhecimento é implementado usando a teoria de autofaces, inicialmente desenvolvida para reconhecimento de faces. Durante a etapa de aprendizado são armazenados em uma base de dados os autovalores e autovetores de várias amostras da imagem em um espaço conhecido como espaço de faces. Para o reconhecimento, uma dada imagem tem seus autovalores e autovetores calculados e estes são comparados um a um com os existentes na base de dados, sendo que o de menor erro associado é dado como o reconhecido, e como resultado obtém-se ainda o erro associado, para que possa ser estabelecido um limite de qualidade mínimo para o reconhecimento [69].

Assim, uma vez reconhecido o objeto na base de dados, calcula-se a posição da plataforma somando à posição conhecida do objeto a distância relativa até a plataforma, eliminando os erros acumulados até o instante pela integração da velocidade do modelo interno.

#### 4.4.5: Integração com o Harpia

O projeto Harpia foi um dos projetos aprovados dentro do edital **CT-INFO 2003 - Software Livre** da FINEP. O projeto visa desenvolver um ambiente gráfico, na concepção de programação aberta, para auxílio na educação, treinamento, implementação e gerenciamento de sistemas de visão [46, 19].

O sistema compõe-se de um conjunto de módulos de programas para o processamento de sinais (predominantemente imagens) e para o gerenciamento remoto de sistemas de visão, fornecendo a possibilidade de construção dos programas para processamento de forma gráfica, como diagramas de blocos, permitindo assim a criação de protótipos de sistemas de visão rapidamente (Figura 54 e Figura 55).

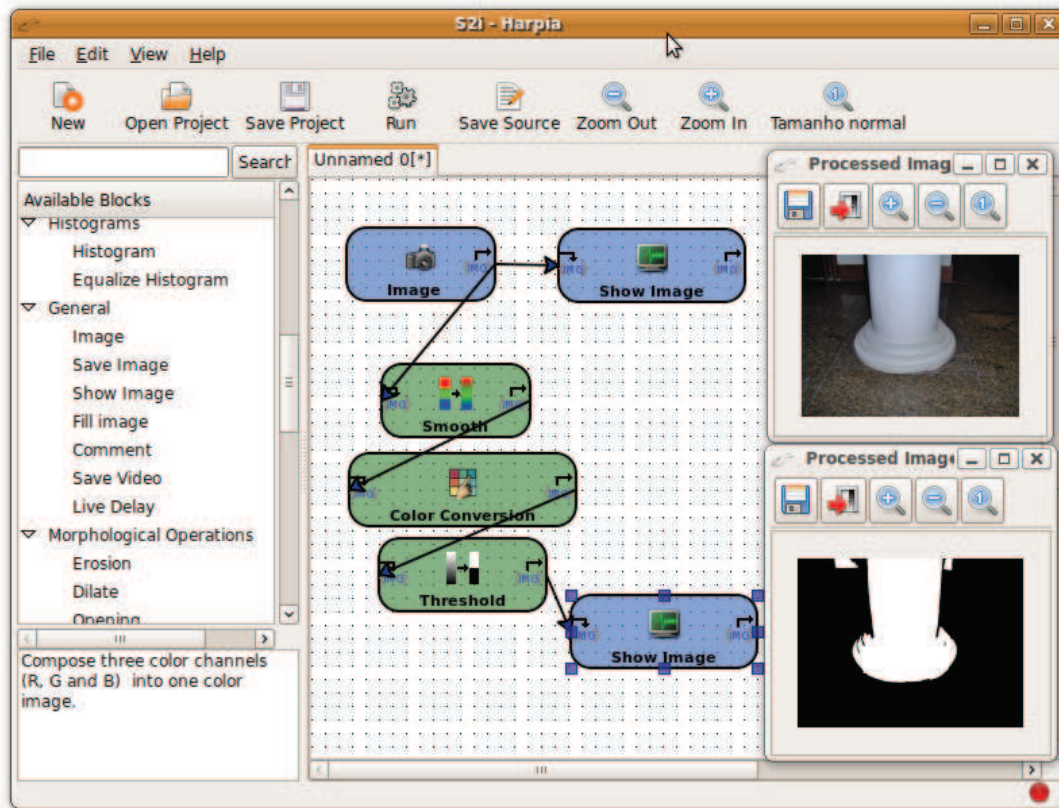


Figura 54: Exemplo de uso do Harpia com uma cadeia de processamento de imagens montada

No decorrer do desenvolvimento da plataforma, foram adicionados ao Harpia as seguintes capacidades:

- Execução de quaisquer comando permitido pelo sistema operacional, com o bloco **Executar Comando**. Este bloco permite comunicação direta com dispositivos, inclusive com as bases dos protótipos anteriores e a plataforma atual.
- Reconhecimento de círculos, quadrados, retas.
- Detecção de faces: as posições de faces na imagem podem ser detectadas.
- Suporte a vários tipos de dados além de imagens: ponto flutuante, pontos em imagem, retângulos.
- Gerar mapas de profundidade por visão estereoscópica.



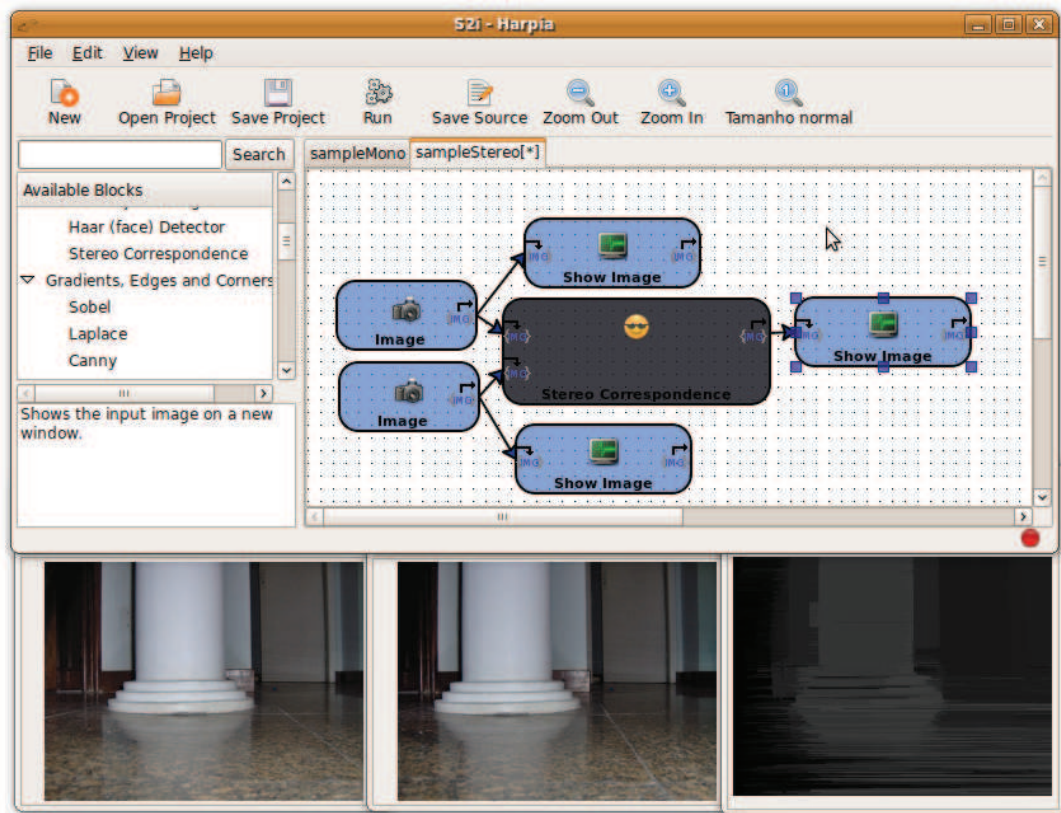


Figura 55: Exemplo de uso com os novos blocos

- Suporte para câmeras múltiplas, arquivos de vídeo e processamento ao vivo.

O desenvolvimento da ferramenta em paralelo permite a execução de testes mais rapidamente e, ao implementar as funcionalidades novas para os trabalhos correntes, a ferramenta vai sendo aperfeiçoada de maneira a abranger um conjunto maior de aplicações.

#### 4.5: Considerações

Neste capítulo foi descrito o trabalho de implementação dos diversos subsistemas envolvidos e também foram apresentados resultados internos de cada subsistema, descrevendo as capacidades de cada um, assim como as limitações encontradas, seja no equipamento, seja na programação e algoritmos disponíveis.

O desempenho de cada elemento encontra-se dentro do esperado, e os testes

realizados para que os resultados parciais fossem adequados possibilitam que, após a integração dos diversos sistemas, consiga-se cumprir com as funcionalidades e objetivos esperados.

O próximo capítulo, **Resultados e Contribuições**, descreve os resultados gerais que puderam ser alcançados após a integração dos diversos sistemas descritos neste capítulo, assim como resultados globais do trabalho.



## 5. Resultados e Contribuições

Neste capítulo serão resumidos os resultados globais obtidos, visto que os resultados parciais dos diversos subsistemas foram descritos no capítulo anterior. Os principais resultados dos sistemas integrados serão apresentados, e são discutidas as contribuições e questões de caráter prático levantadas durante o projeto.

### 5.1: Projetos técnica e financeiramente viáveis

Apesar do caráter experimental e acadêmico, foi possível concluir o projeto completo de um robô móvel com rodas (Figura 56), com processamento de alto nível (plataforma PC i386) e possibilidade de expansões, com custo abaixo de US\$ 1000 ( R\$ 2500 com impostos de importação e frete de componentes importados). O Apêndice C apresenta uma lista de componentes, peças, materiais utilizados e o custo aproximado para eles no período de desenvolvimento (anos de 2007-2008).

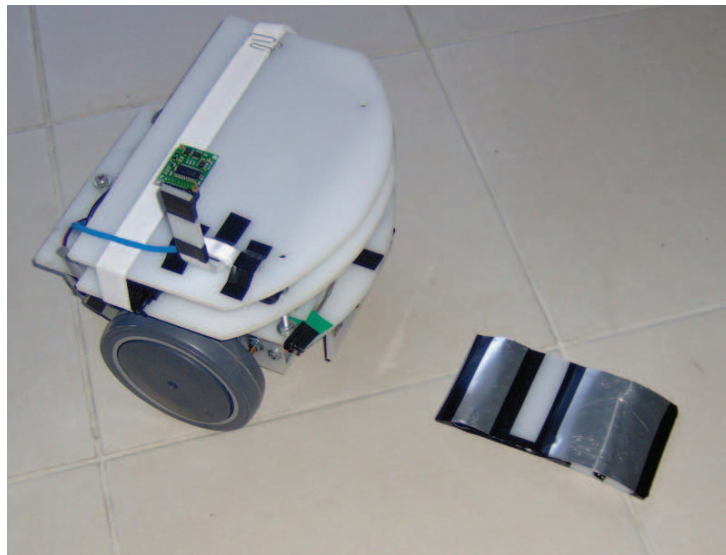


Figura 56: Plataforma completa e funcional, com sistema de visão estereoscópica

Com exceção dos sensores (bússola, acelerômetro, medidor de distância), todas as outras peças podem ser (e foram) adquiridas no Brasil no mercado para consumi-

dor final. Assim, por fim obteve-se uma plataforma que pode ser utilizada como um bom começo para pesquisa na área de robótica móvel com rodas, substituindo soluções equivalentes por uma pequena parcela do custo e possibilitando a nacionalização da tecnologia empregada.

A plataforma apresenta ainda o uso de componentes modernos que ainda não foram inteiramente absorvidos no mercado atual de robótica móvel, mas que tem futuro iminente no ramo (acelerômetros, bússola, microcontroladores facilmente reprogramáveis e sistemas i386 completos compactos a baixo custo). Algumas destes componentes já fazem parte de produtos disponibilizados durante o curso do desenvolvimento, ilustrando a atualidade da pesquisa.

Realizou-se ainda o projeto de um circuito de potência para acionamento dos motores seguro, eficiente e de baixo custo. Os circuitos de potência são críticos para o desenvolvimento de atuadores e sua fácil disponibilidade e baixo custo permitem que potencialmente se desenvolvam ainda outras aplicações. Assim, foram utilizados componentes eletrônicos fáceis de serem encontrados, enquanto que, por já estar sendo utilizado no grupo Robota a cerca de 3 anos, acabou-se por formalizar um projeto já testado e comprovado.

O circuito de potência é seguro no sentido em que é auto-protegido de forma a permitir o acionamento do motor no sentido direto e reverso ao mesmo tempo ocasionando o travamento do motor, sem que isso cause problemas elétricos ou colapso dos sistemas. Quando comparado aos circuitos de baixo custo tradicionais, o sistema gera menos calor e apresenta menos perdas, com uma margem grande de operação, com capacidade original de 5A, até 50VDC, podendo facilmente ser expandido para até 10A.

## **5.2: Substituição de realimentação interna por realimentações externas**

Enquanto o sensoriamento das variáveis internas é mais fácil de ser realizado (sensores baratos, ambiente mais controlado), é possível que os dados não sejam representativos quando deseja-se que o robô interaja com o meio. Assim, a realimentação utilizando dados externos leva a interação robô/ambiente a um novo nível, pois permite que as variáveis sobre as quais o robô deseja operar sejam diretamente observadas ao

invés de estimadas.

Neste sentido, ilustra-se o sucesso do controle de ângulo de robôs móveis usando realimentação externa e medidas absolutas via bússola. Apesar das não linearidades do sistema, foi possível projetar um controle capaz de operar na maioria das ocasiões (variações de carga, piso irregular, e diferentes estados da bateria) e, quando tal não for possível, ainda é possível detectar o erro.

Além de permitir um controle mais fino da direção desejada, para o mapeamento, a medição direta da orientação elimina grande parte dos erros. Por outro lado, se forem utilizados somente métodos diferenciais para medição de ângulo (como integração da componente de rotação devido à ação diferencial das rodas), acaba-se gerando um erro crescente que, amplificado pela distância percorrida (efeito alavanca) podem tornam os mapas completamente inválidos. Como a medida da bússola é absoluta, os erros, mesmo que ocorram, são momentâneos e afetam apenas localmente o mapa.

Como esse tipo de sensor é bastante sensível a alterações no campo magnético e o conjunto motores/bateria/cabos energizados afetam as medições, um procedimento de calibração automático foi implementado, de forma que fica fácil alterar o posicionamento da bússola na estrutura mecânica e executar a recalibração para obter medições adequadas.

Além da bússola, é possível a realimentação externa de aceleração via acelerômetro. Apesar da exatidão fornecida por este tipo de sensor ainda não ser muito satisfatória, ainda assim é possível comparar a variação de velocidade comandada para o motor com a aceleração medida pelo acelerômetro. Assim, são detectadas situações anormais, como colisão, obstrução, ou derrapagem indesejada e assim corrigir esses problemas (incluindo os resultados no modelo interno que estima a velocidade, por exemplo).

Caso não possa utilizar as referências externas de aceleração, o robô tem que se basear somente na informação interna para alimentar o modelo de mundo e, consequentemente, não se torna muito robusto em ambientes dinâmicos.

Por último, a medição da aceleração funciona como um sensor de colisão tátil.

Se qualquer parte do robô for mexida sem que ele próprio tenha enviado o comando, esse movimento é registrado e ações podem ser tomadas no sentido de tentar garantir a segurança da plataforma fazendo com que os motores sejam desligados.

### **5.3: Confiabilidade com sensores adicionais**

O uso de sensores para medição de distância permite que, mesmo na ausência de luminosidade ou mal-funcionamento dos sistemas de visão, a plataforma possa ainda operar. Os sensores escolhidos para medição de distância são muito confiáveis, no entanto o preço é mantido baixo devido ao encapsulamento mecanicamente menos resistente que os sensores industriais equivalentes.

A curva de calibração foi levantada para os sensores de distância e o ajuste das medidas é realizado diretamente no microcontrolador. Comprovou-se ainda que, dentre sensores do mesmo modelo, o uso de curvas de calibração iguais leva a bons resultados.

### **5.4: Facilidade para expansão e reprogramação**

As capacidades do microcontrolador escolhido permitem futuras expansões, ao mesmo tempo em que permite fácil alteração em sua programação, devido à linguagem padronizada ANSI-C e a possibilidade de gravação dos novos programas pela mesma interface USB com a qual o sistema se comunica, sem necessidade de circuitos de programação adicionais.

O microcontrolador se comunica via interface USB, comum em computadores modernos, mas pode se comunicar via RS-232 ou inclusive é possível alterar a programação do microcontrolador de forma que a plataforma opere utilizando-o como subsistema computacional, além das funções acumuladas do subsistema eletro-eletrônico.

O uso de um PC ultra-móvel, que se baseia em uma arquitetura tradicional, facilita o desenvolvimento e depuração dos programas, e o uso da interface USB torna o sistema como um todo flexível a futuras modificações, podendo todo o subsistema computacional ser substituído por equivalentes mais potentes e modernos com facilidade.

Além disto, o subsistema computacional é constituído de um computador com-

pleto, com rede sem-fio IEEE 802.11, visor LCD, sistema de entrada e saída de áudio, armazenamento em disco de estado sólido, três portas USB para conexão de periféricos e uma câmera embutida que pode ser utilizada para realizar testes com visão computacional, facilitando ainda mais intervenções diretas na plataforma e envio de informações utilizando rede sem fio com um protocolo comum e padronizado.

As duas câmeras digitais adicionadas (com sensor CMOS 1.3MP) são compactas e têm lentes de baixa distorção, sendo mecanicamente alinhadas para visão estereoscópica, também utilizando a convencional interface USB.

### **5.5: Programação completa da estrutura básica para controle de alto nível da plataforma**

Foram experimentadas várias formas de comunicação para interfaces homem / máquina: Interface gráfica (local/remota via SSH), rede via *sockets*, rede via protocolo XMPP, com acesso fácil via Internet e navegador. As interfaces, além de possibilitar a interação do usuário com a máquina, permitem ainda que exista comunicação de máquina a máquina, ao utilizar padrões conhecidos.

Protocolos recentes, como o XMPP, eliminam boa parte da gerência de conexão, sendo possível que o robô torne-se um servidor que forneça serviços via Internet sem necessidade alguma de configurações especiais para tal, diferentemente do que ocorreria com servidores via *sockets*, uma vez que grande parte da complexidade está no agente de autenticação XMPP, que pode ser redundante e descentralizado.

O programa principal da plataforma é executado com múltiplos fluxos de execução, que torna o sistema pronto para execução em processadores de múltiplos núcleos. Além disso, a fatia de uso do processador pode ser individualmente regulada modificando o período das tarefas, aumentando o período para liberar mais tempo de processador para outras tarefas e diminuindo para melhorar a velocidade de resposta do robô às alterações do ambiente.

Dentro do campo de sistemas operacionais, o escalonamento de tarefas (periódicas) com cargas de processamento variável é um tópico de pesquisa importante e bastante estudado. A plataforma permite que pesquisas como essa possam ser experi-

mentadas em uma aplicação real.

Realizaram-se ainda medições de latência para núcleo com suporte a tempo real, podendo-se assim reduzir as latências máximas para 15ms, em detrimento da confiabilidade de comunicação utilizando protocolo USB, que é, de fato, não determinístico.

Do ponto de vista de capacidades da plataforma propriamente dita, foram implementados algoritmos de controle de posição, mapeamento vetorial, navegação com geração de trajetórias utilizando busca em grafos e evitamento de obstáculos, utilizando soluções típicas descritas na literatura, comprovando os resultados esperados e avaliando as limitações impostas pelas simplificações utilizadas.

### **5.6: Visão (estereoscópica) embarcada com aplicações já testadas**

Os sistemas de visão implementados incluem medição de distância pelos métodos monocular e estéreo, além de propor a fusão de medidas de distância e estudar métodos mais aprimorados de fusão de sensores.

Os procedimentos de calibração das câmeras implementados por Maurício Stivanello também foram incluídos, assim como os resultados de seu trabalho puderam ser postos em prática, utilizando as técnicas para redução de erros de disparidade e modificações propostas para melhoria do desempenho [61, 62].

Assim, foram implementados modos distintos para visão estereoscópica chaveada, fornecendo qualidade para serviços finos, com a medição mais exata de apenas uma correspondência entre pontos e velocidade para serviços rápidos, com correspondência densa (Figura 57).

Os algoritmos de correspondência densa velozes são usados para obter informação rápida e grosseira de vários pontos, e quando é necessário medir com mais exatidão podem ser utilizados algoritmos de correspondência pesados que não assumem alinhamento, podendo os modos serem chaveados durante a operação normal, incluindo alterações necessárias no período do fluxo de execução da câmera [61, 62].

Além dos algoritmos de correspondência densa já implementados por Stivanello [61, 62], foram implementados alguns algoritmos de correspondência esparsa que me-

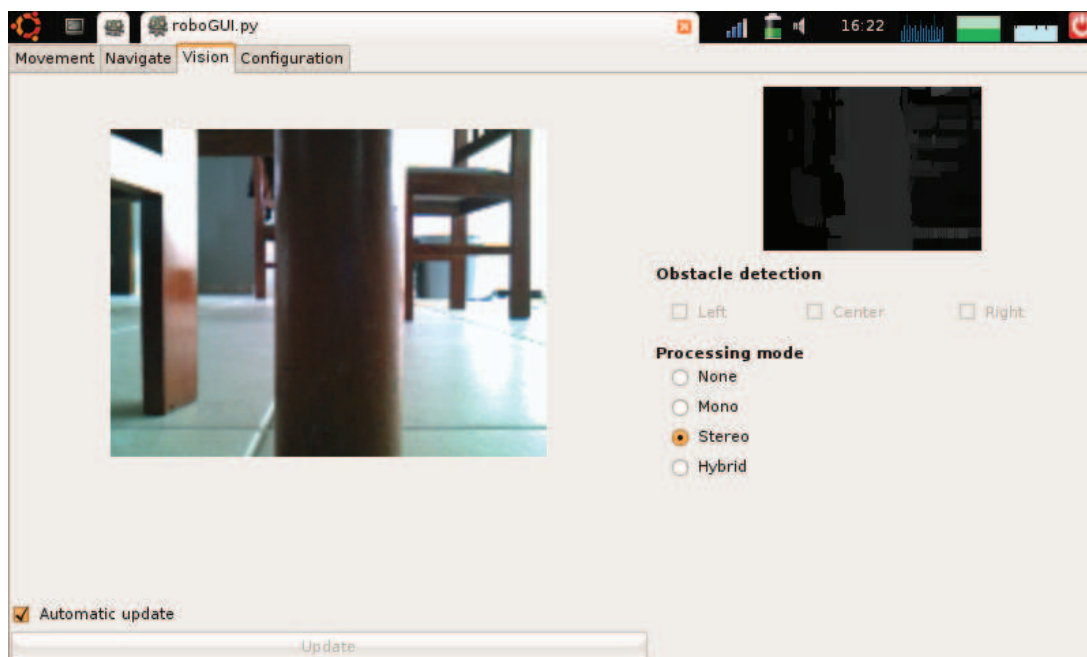


Figura 57: Interface gráfica mostrando os resultados da visão estereoscópica no modo denso

dem pontos específicos com exatidão maior (utilizando imagens de maior resolução, aceitando falhas de alinhamento e algoritmos de pareamento mais complexos), bons para medir distância de objetos cuja posição é conhecida na imagem, por exemplo (Figura 58).

Como resultado final, com todos os sistemas operando simultaneamente, executou-se o mapeamento da sala observada na Figura 59, obtendo os mapas que podem ser vistos nas Figs. 60 e 61. O mapa construído somente com sensores de proximidade levou cerca de 5 minutos e 30 segundos para ser completado, ao fim dos quais a plataforma chegou a conclusão de que estava em um quarto fechado.

Utilizando sistemas de visão o robô chegou a um mapa simplificado, que em cerca de 1 minuto e 30 segundos permitiu a máquina concluir, da mesma forma, que estava em um quarto fechado. O mapa simplificado apresenta ainda menos medições espúrias e sintetiza bem o ambiente navegado.

Neste exemplo testado, o uso da visão estereoscópica foi essencial, pois as paredes apresentavam uma tonalidade bastante semelhante ao chão, necessitando de todos





Figura 58: Resultados de correspondência esparsa para medição de distância de objeto (estojo preto). A distância (em pontos) é 6 pontos, que pode ser convertida em distância métrica posteriormente.

os detalhes possíveis para que as paredes fossem detectadas pelo sistema de visão.



Figura 59: Quarto retangular com mobília (ambiente semi-estruturado), vista de topo e fotos de três ângulo.

Foram implementados ainda algoritmos de localização que fornecem um ponto de



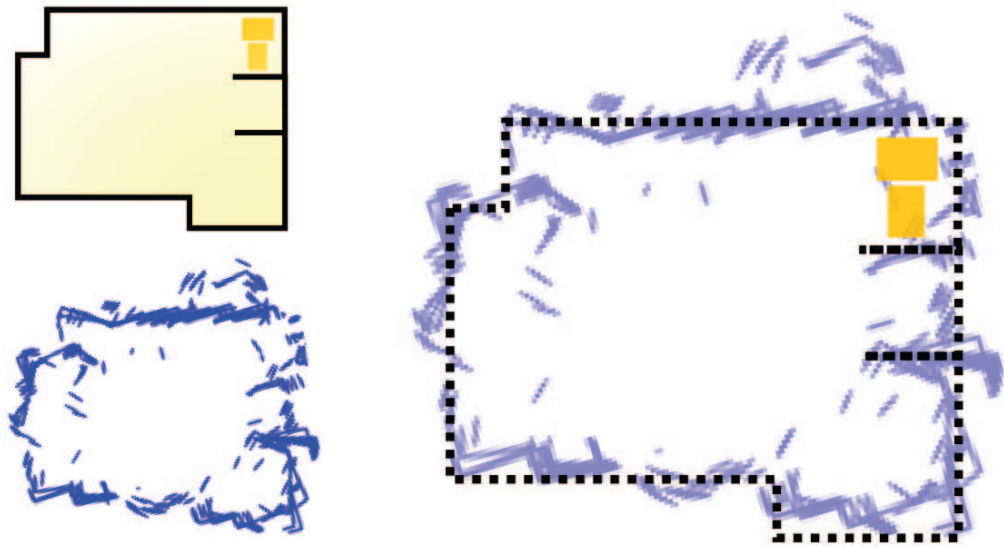


Figura 60: Quarto retangular com mobília mapeado pelo robô usando apenas sensores de proximidade

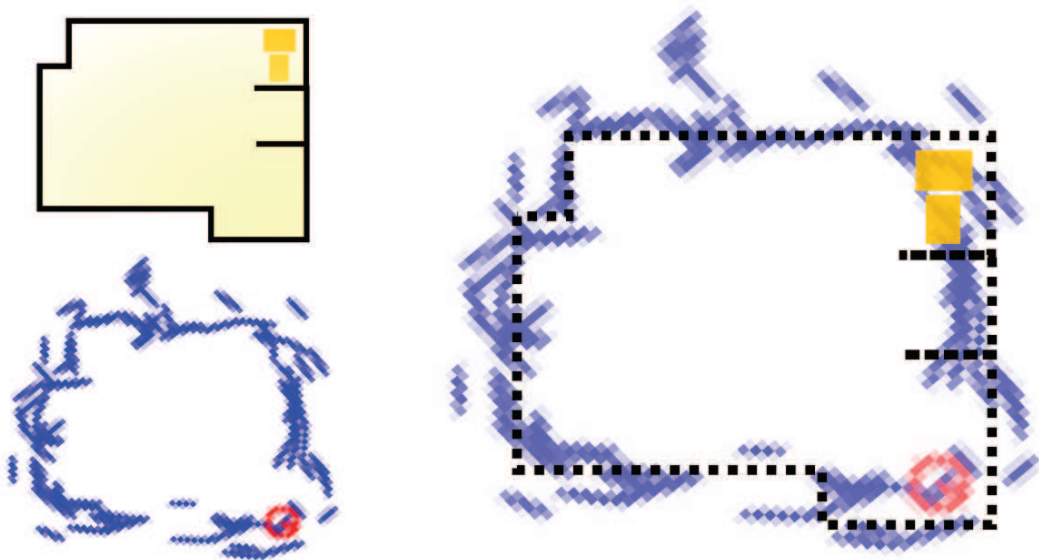


Figura 61: Quarto retangular com mobília mapeado pelo robô usando visão computacional

partida para o estudo de reconhecimento de objetos visando a localização por marcos ou para manipulação com módulos adicionais com atuadores instalados na plataforma.

### **5.7: Flexibilidade para alterações drásticas, seguindo a mesma concepção básica**

A concepção inicial da arquitetura permite que módulos sejam substituídos mantendo a interoperabilidade. Por exemplo, o subsistema eletro-eletrônico baseado em microcontrolador Arduino pode ser substituído pelo baseado em Microchip 18F24550, como ocorre, por exemplo, no projeto do robô Robota-WiDe (2008).

Além disto, o sistema computacional pode ser utilizado com uma base simulada, com o subsistema eletro-eletrônico sendo simulado internamente, permitindo testes sem a necessidade da plataforma física.

Em um projeto posterior, utilizando a mesma concepção da plataforma projetada, o grupo Robota construiu um sistema composto de dois robôs cooperando em um sistema mestre-escravo, compartilhando o mesmo subsistema computacional (chamado robô Robota-WiDe).

Um dos robôs (chamado de escravo) apresenta visão monocular sobre uma base rotativa, no qual foi implementado um sistema realimentado de controle de orientação de câmera para manutenção do alinhamento da câmera com o objeto, mesmo que o robô seja movimentado. A Figura 62 ilustra o controle de orientação da câmera, mostrando o alinhamento sendo concluído cerca de 1,3s após o início da movimentação do robô, que executava uma mudança de orientação de cerca de 60°.

Adicionalmente, neste mesmo robô escravo, foram implementados dois sistemas de medição de distância até o objeto por visão monocular, e os resultados de ambos os sistemas são fundidos com a utilização do filtro de Kalman, comprovando os resultados esperados para sua utilização.

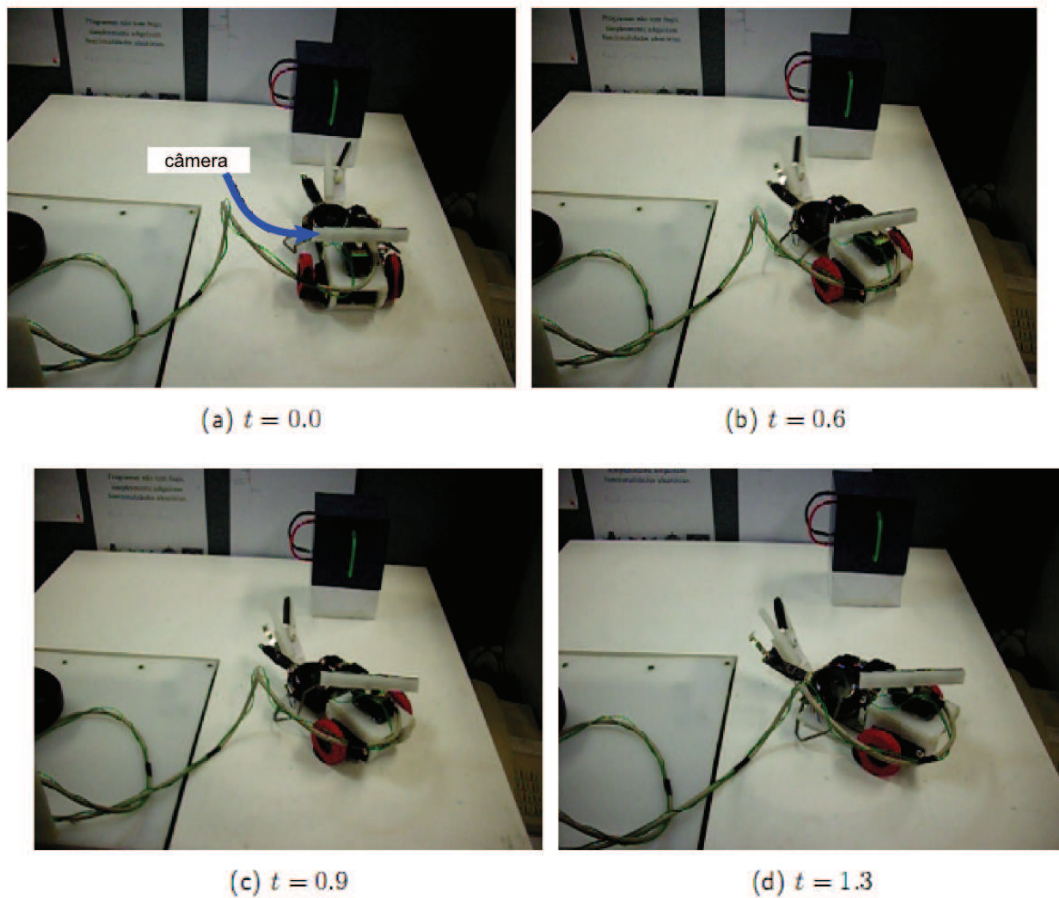


Figura 62: Controle de ângulo do robô/câmera para alinhamento com bloco

### 5.8: Considerações

A estrutura proposta para delimitação dos diversos subsistemas possibilita uma separação dos problemas, assim mais facilmente resolvidos, sendo que a integração dos diversos sistemas pode ser realizada facilmente quando as interfaces são bem definidas. Os resultados globais desta abordagem puderam ser conferidos neste capítulo.

Além disso, a implementação da plataforma permitiu que se realizassem os testes com visão estéreo em navegação para robótica móvel, mas além disso, no decorrer do projeto várias linhas interessantes de pesquisas futuras surgem, como, por exemplo, a interação entre homem-máquina, o controle da cinemática de robôs móveis levando em conta as diversas não linearidades encontradas, técnicas de fusão de sensores incluindo sensores de alto-nível (como câmeras) e sensoriamento baseado em realimentação de

variáveis externas.

É possível o desenvolvimento de projetos de novos robôs, dentro da mesma estrutura, mas com objetivos totalmente distintos, como foi já experimentado com um protótipo novo (Robota-WiDe) incluindo dois robôs cooperando em um sistema mestre-escravo.

## 6. Conclusão e perspectivas

À medida que o mercado vai se popularizando e os custos de desenvolvimento diminuindo, a tendência natural é que a disponibilidade de robôs móveis, seja para pesquisa ou para o consumidor final, aumente.

Neste cenário que se propôs a criação de uma plataforma própria para os testes. Parte do esforço de projeto se justifica pelas necessidades únicas da pesquisa com sistemas de visão intimamente integrados com os demais sistemas.

No entanto, mais do que isso, o projeto e implementação de um sistema próprio fornecem total liberdade na alteração do projeto, assim como permite que surjam pesquisas partindo desde o elemento mais específico e interno do projeto, como o controle da potência dos motores, até as aplicações gerais que poderiam também ser criadas com o uso de plataformas comerciais, como sistemas de mapeamento com sensores de distância.

Como tecnologia, o robô apresenta o uso de sensores modernos para realimentação externa, como bússola e acelerômetros. Como trabalho de engenharia, custos foram mantidos baixos e o projeto modularizado foi levado adiante como forma de dividir problemas grandes e resolvê-los.

Como pesquisa, um maior esforço foi empenhado nos sistemas de visão, modificando sempre que necessário os métodos existentes para obter o desempenho desejado e resultados válidos. Além disso, foi formalizado o projeto do circuito de potência para acionamento de motores, projetado o controlador de orientação realimentado pelas leituras da bússola.

Por fim, no decorrer do trabalho, muitas soluções típicas da literatura foram empregadas, com resultados parecidos com os demonstrados nos trabalhos originais, de forma consistente e reprodutível.

O estado da arte mostra a necessidade de um robô aberto, com flexibilidade e extensibilidade, de baixo custo baseado em padrões vigentes. A avaliação experimental

de métodos de localização, mapeamento e navegação permitiram validar a plataforma proposta. Por fim, tem-se uma plataforma robótica móvel para testes de sistemas de visão embarcados flexível e de baixo custo.

Algumas soluções não tão convencionais, ou não clássicas, como filtro de Kalman para fusão de sensores e projeção no espaço de autofaces para reconhecimento de objetos, foram experimentadas em ambiente controlado, sem serem embarcados no robô, com relativo êxito.

No entanto, ainda resta o trabalho de generalizar as aplicações testadas para melhor integração no sistema atual: o filtro de Kalman precisa ser mais estudado em um protótipo posterior e o reconhecimento de objetos ainda é restrito a objetos sintéticos, sendo que o processo de separação de objetos de uma cena ainda é um problema complexo e não resolvido.

As possibilidades de expansão e alteração da plataforma são grandes, e entre as possíveis modificações, estão sendo estudadas algumas no sentido de adicionar funcionalidades como reconhecimento de fala, sintetização de fala, adição de ferramentas de manipulação, sistemas de visão ativa com luz estruturada, o uso de visão como realidade externa de deslocamento, permitindo detecção de deslizamentos.

# Apêndices

## A. Equipes de desenvolvimento dos protótipos analisados

Protótipo	Equipe	Observação
Ajax	Alexandre Dotto Christian E. M. Silvano Daniel C. Kulkamp Mathias J. K. Erdtmann	Desenvolvido para o LARC2005-IEEE
Antioch	Alexandre Dotto Clovis P. Scotti Eduardo Otte Hülse	Desenvolvido para o LARC2006-IEEE
RF	Roni Rigoni	
El Cabron	Clovis P. Scotti Eduardo Otte Hülse Renata Faraco Cunha Tiê Teixeira Lima Penatti Mathias J. K. Erdtmann	Desenvolvido para o LARC2007-IEEE
WiDe*	Clovis P. Scotti Gabriel Felipe Lehnhard Rodrigo Donadel Roni Rigoni Mathias J. K. Erdtmann	Desenvolvido para o LARC2008-IEEE

\* - Inclui os trabalhos com filtro de Kalman e alinhamento camera/objeto.





## B. Opções de plataformas robóticas disponíveis no mercado

Nome	Fabricante	Arq.	CPU	Câmera	Sens.	Preço
Pioneer	MobileRobots Inc.	(F)	2	mono (+\$2K) estéreo(+ \$7K)	toque(+\$.5K) bússola(+ \$1.4K)	\$12K
PC-BOT914	White Box Rob.	(A)	3	mono (inc)	prox.(-)	\$9K
Pekeell	Wany Robotics	(F)	2	mono (+\$.3K)	prox.(-) bússola(+\$.3K) aceler.(+\$.3K)	\$7.3K
SRV-1 Blackfin	Surveyor Corp.	(A)	1	mono (-)	(-)	\$.5K
<b>Proposta deste trabalho</b>	DAS-UFSC	(A)	2	mono(-) estéreo(-)	aceler.(-) bússola(-) prox.(-)	\$1K (custo)

Nota: Valores em US\$ (\$1K = US\$1.000,00), excluindo impostos e frete, para universidades (quando disponível preços reduzidos para universidade estes foram utilizados). O valor adicional escrito após o acessório é o valor cobrado para adicioná-lo ao robô e (inc) significa incluso no preço.

Legenda:

Arq. - Arquitetura: (F) fechada (A) aberta.

CPU - Poder de processamento, escala de 1 a 3, onde 1 equivale a um processador de baixo consumo de energia de 300MHz e 3 a um processador de 2 núcleos, 2GHz.

Sens. - Sensores: prox. = proximidade; aceler. = acelerômetro.



## C. BOM (*Bill Of Materials*) Lista de materiais para fabricação

Subsistema	Item	US\$	R\$
Mecânico	2 Motorredutores Bosch (vidro elétrico automotivo) Torque 10 Kg.cm 100RPM @12V	45,00	90,00
	2 Rodas industriais 6"	10,00	20,00
	1 Roda industrial livre 3"	4,00	8,00
	Serviço de soldagem de eixos	7,50	15,00
	Fuso galvanizado (1m)	1,50	3,00
	Porcas, arruelas, parafusos	5,00	10,00
	8 L gavanizado 3cm	2,00	4,00
	Placas PP 4xA4	75,00	150,00
Subtotal		150,00	300,00
Elétrica	Bateria 7.2Ah 12V, Chumbo-Ácido	20,00	40,00
	Cabos/fios	5,00	10,00
	Conectores Molex	5,00	10,00
	Placa universal	1,00	2,00
	2 PonteH H5A (fabricação própria)	10,00	20,00
	Carregador bateria Chumbo-Ácido	41,00	82,00
Eletrônica	(FE) Arduino Nano	60,00	120,00
	(I) Bússola Devantech CMPS3	80,00	160,00
	(I) 3 Sensores de distância Sharp GP2D120	65,00	130,00
	(I) Acelerômetro 2 eixos 2g DimensionEng. ADXL322	45,00	90,00
Subtotal		332,00	664,00
Computacional	(FE) PC ultra-móvel EeePC 701 ( <i>subnotebook</i> )	450,00	900,00
	(FE) 2 Câmeras HP CMOS 1.3MP (peças de reposição para <i>notebooks</i> HP)	50,00	100,00
Subtotal		500,00	1000,00
<b>TOTAL</b>		<b>982,00</b>	<b>1964,00</b>

Notas:

- Valores aproximados para setembro de 2008 (cotação usada para conversão US\$1,00 = R\$2,00).

- O produto cujo nome iniciar em (I) foi importado.

- O produto cujo nome iniciar em (FE) é de fabricação estrangeira.



## Referências

- [1] CFS scheduler design - Ingo Molnár (<http://people.redhat.com/mingo/cfs-scheduler/sched-design-cfs.txt>). Acessado em 10 de julho de 2007.
- [2] Linux: Completely fair scheduler merged (<http://kerneltrap.org/node/11737>). Acessado em 10 de julho de 2007.
- [3] Sítio da ASIMO - Honda Motor Co. (<http://world.honda.com/asimo/>). Acessado em 9 de fevereiro de 2009.
- [4] Sítio da base PPR (CMU) (<http://www.cs.cmu.edu/~reshko/pilot/>). Acessado em 15 de abril de 2009.
- [5] Sítio da Devantech Ltd. (<http://www.robot-electronics.co.uk>). Acessado em 20 de novembro de 2008.
- [6] Sítio da Dimension Engineering (<http://www.dimensionengineering.com/index.html>). Acessado em 20 de novembro de 2009.
- [7] Sítio da Hitec Inc. (<http://www.hitecrobotics.com/>). Acessado em 9 de fevereiro de 2009.
- [8] Sítio da Infinuvo Corporation ([infinuvo.com](http://infinuvo.com)). Acessado em 3 de agosto de 2008.
- [9] Sítio da iRobot Corporation (<http://irobot.com>). Acessado em 3 de agosto de 2008.
- [10] Sítio da Lynxmotion Inc. (<http://www.lynxmotion.com/>). Acessado em 15 de abril de 2009.
- [11] Sítio da Microchip Technology Inc. (<http://www.microchip.com/>). Acessado em 20 de novembro de 2008.
- [12] Sítio da MobileRobots Inc. (<http://www.activrobots.com>). Acessado em 3 de agosto de 2008.

- [13] Sítio da Python Software Foundation (<http://python.org/>). Acessado em 7 de dezembro de 2008.
- [14] Sítio da Sharp Electronics Corporation (<http://www.sharpsma.com>). Acessado em 20 de novembro de 2008.
- [15] Sítio da Surveyor Corporation (<http://www.surveyor.com>). Acessado em 15 de outubro de 2008.
- [16] Sítio da Wany Robotics (<http://www.wanyrobotics.com>). Acessado em 15 de outubro de 2008.
- [17] Sítio da White Box Robotics (<http://www.whiteboxrobotics.com/>). Acessado em 3 de agosto de 2008.
- [18] Sítio do EyeBot (<http://www.joker-robotics.com/eyebot/socbot.html>). Acessado em 15 de abril de 2009.
- [19] Sítio do Projeto Harpia - S2i (<http://s2i.das.ufsc.br/harpia>). Acessado em 20 de fevereiro de 2009.
- [20] Sítio do Robota - DAS - UFSC (<http://robota.das.ufsc.br/>). Acessado em 20 de fevereiro de 2009.
- [21] Sítio do S2i - Sistemas Industriais Inteligentes - DAS - UFSC (<http://s2i.das.ufsc.br/>). Acessado em 20 de fevereiro de 2009.
- [22] Sítio oficial do Arduino Nano (<http://arduino.cc/en/main/arduinoboardnano>). Acessado em 20 de novembro de 2008.
- [23] BACHNAK, R., AND ESPARZA, J. *Remotely Operated Vessel for Environmental Studies in Shallow Water Areas*, vol. 1. North Atlantic University Union, London - UK, 2007.
- [24] BHALLA, K., DURDLE, N. G., PETERSON, A. E., RASO, J., HILL, D., AND LI, X. Automatic feature detection and correspondence in a stereo-vision application. In *Intelligent Systems for the 21st Century*, vol. 4, pp. 3537–3542. Vancouver, 1995.

- [25] BHUSNURMATH, A., AND TAYLOR, C. Solving stereo matching problems using interior point methods. *3D Data Processing, Visualization and Transmission*. Atlanta, GA, USA, 2008.
- [26] BIRCHFIELD, S., AND TOMASI, C. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision* 35, 1073–1080. Hingham, MA, USA, 1999.
- [27] BLAKE, A., AND YUILLE, A. *Active Vision*. MIT Press. Cambridge - USA, 1992.
- [28] BLEYER, M., AND GELAUTZ, M. A layered stereo algorithm using image segmentation and global visibility constraints. *International Conference on Image Processing* 5. Singapura, 2004.
- [29] DE FRANÇA, J. A., DE FRANÇA, M. B., AND STEMMER, M. R. Stereo-based detection and localization of obstacles in indoor environments. *VI Induscon*. Joinville - Brasil, 2004.
- [30] DELAGE, E., LEE, H., AND NG, A. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2418–2428. Washington - USA, 2006.
- [31] DESOUSA, G. N., AND KAK, A. C. Vision for mobile robot navigation: A survey. *IEEE Transactions on pattern analysis and machine intelligence* 24, 2. Piscataway-EUA, 2002.
- [32] DOUAT, L. R., CASTELAN, E. B., AND MORENO, U. F. Stabilization of a 5-link bipedal robot by means of dorsal movement compensation. In *13th IEEE International Conference on Emerging Technologies and Factory Automation* (2008), IEEE, pp. 1092–1095. Hamburg - Alemanha, 2008.
- [33] ERDTMANN, M. J. K., SILVANO, C. E. M., AND STEMMER, M. R. Modelo de visão computacional de baixo nível antropomórfica com aplicação em robótica móvel. *SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing* 2005 1, 1. Natal - Brasil, 2005.

- [34] FERGUSON, D., AND STENTZ, A. Anytime rrts. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing, China, 2006.
- [35] GAO, L. F., GAI, Y.-X., AND FU, S. Simultaneous localization and mapping for autonomous mobile robots using binocular stereo vision system. *IEEE International Conference on Mechatronics and Automation*. Harbin - China, 2007.
- [36] GEORGOULAS, C., KOTOULAS, L., SIRAKOULIS, G. C., ANDREADIS, I., AND GASTERATOS, A. Real-time disparity map computation module. *Microprocess. Microsyst.* 32, 3, 159–170. Amsterdam - Holanda, 2008.
- [37] GONZALEZ, R. C., AND WOODS, R. E. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.
- [38] HARTLEY, R., AND ZISSERMAN, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, Massachusetts, 2004.
- [39] HEMA, C. R., PAULRAJ, M. P., NAGARAJAN, R., AND SAZALI, Y. Object localization using stereo sensors for adept scara robot. *IEEE Conference on Robotics, Automation and Mechatronics*. Bangkok - Tailândia, 2006.
- [40] HOIEM, D., EFROS, A. A., AND HEBERT, M. Geometric context from a single image. In *International Conference of Computer Vision (ICCV)*, vol. 1, Institute of Electrical and Electronics Engineers, pp. 654 – 661. Beijing - China, 2005.
- [41] HOOPES, D., DAVIS, T., NORMAN, K., AND HELPS, R. An autonomous mobile robot development platform for teaching a graduate level mechatronics course. *Frontiers in Education, Annual 2*, F4E–22. Los Alamitos - USA, 2003.
- [42] HRABAR, S. E., CORKE, P. I., SUKHATME, G. S., USHER, K., AND ROBERTS, J. M. Combined optic-flow and stereo-based navigation of urban canyons for a UAV. *Intelligent Robots and Systems 2005*, 3309–3316. Piscataway - USA, 2005.
- [43] HU, W., DOWNS, T., WYETH, G., MILFORD, M., AND PRASSER, D. A modified particle filter for simultaneous robot localizarion and landmark tracking in an indoor environment. In *In Proceedings of the Australasian Conference on Robotics and Automation*. Canberra - Austrália, 2004.



- [44] HUHA, K., PARK, J., HWANG, J., AND HONG, D. A stereo vision-based obstacle detection system in vehicles. *Optics and Lasers in Engineering* 46. Maryland Heights - USA, 2008.
- [45] JENNINGS, C., AND MURRAY, D. Stereo vision based mapping and navigation for mobile robots. In *Proc. IEEE International Conference on Robotics and Automation*, 1694–1699. Albuquerque - EUA, 1997.
- [46] JUNGES, L. C. D., ERDTMANN, M. J. K., AND STEMMER, M. R. Desenvolvimento de uma ferramenta para auxílio na educação, treinamento, implementação e gerenciamento de sistemas de visão. *SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing 2007* 1, 1. Belo Horizonte - Brasil, 2007.
- [47] LIM, H., AND BINFORD, T. Stereo correspondence: A hierarchical approach. *Image Understanding Workshop* 87, 234–241. Los Angeles - USA, 1987.
- [48] MARTINS, N. A., BERTOL, D., DE PIERI, E. R., AND CASTELAN, E. B. Control of mobile robot considering actuator dynamics with uncertainties in the kinematic and dynamic models. In *10th International Work-Conference on Artificial Neural Networks*, vol. 5518, Springer, pp. 1256–1263. Salamanca - Espanha, 2009.
- [49] MEDIONI, G., FRANÇOIS, A. R., SIDDIQUI, M., KIM, K., AND YOON, H. Robust real-time vision for a personal service robot. *Computer Vision and Image Understanding*, 108. Orlando - USA, 2007.
- [50] MICHELS, J., SAXENA, A., AND NG, A. Y. High speed obstacle avoidance using monocular vision and reinforcement learning. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, ACM, pp. 593–600. New York - USA, 2005.
- [51] NILSSON, N. J. Shakey the robot. *Techreport*, 323. Menlo Park - USA, 1984.
- [52] OWENS, K., AND MATTHIES, L. Passive night vision sensor comparison for unmanned ground vehicle stereo vision navigation. In *Proc. IEEE International Conference on Robotics and Automation*. San Francisco - EUA, 2000.

- [53] QUEK, B. K., IBAFIEZ-GUZMÁN, J., AND LIM, K. W. Feature detection for stereo-vision-based unmanned navigation. *IEEE Conference on Cybernetics and Intelligent Systems*. Singapura, 2004.
- [54] SAXENA, A., CHUNG, S. H., AND NG, A. Y. 3-d depth reconstruction from a single still image. *International Journal of Computer Vision* 76, 2007. Amsterdam - Holanda, 2007.
- [55] SAXENA, A., CHUNG, S. H., AND NG, A. Y. Learning depth from monocular images. *Neural Information Processing Systems*, 18. Vancouver - Canadá, 2005.
- [56] SAXENA, A., SCHULTE, J., AND NG, A. Y. Depth estimation using monocular and stereo cues. *International Joint Conference on Artificial Intelligence*. Amsterdam - Holanda, 2007.
- [57] SCHARSTEIN, D., AND SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision* 47, 7–42. Hingham - USA, 2002.
- [58] SCOTTI, C. P., ERDTMANN, M. J. K., AND STEMMER, M. R. Pose control in mobile robots. *Simpósio Brasileiro de Automação Inteligente*. Florianópolis - Brasil, 2007.
- [59] SIEGWART, R., AND NOURBAKHSH, I. R. *Introduction to Autonomous Mobile Robots*. MIT Press, Cambridge, Massachusetts, 2004.
- [60] SILVA, F. A. E., STRASSMANN, D. S., DA COSTA, J. G. F., LOSSIO, R. G., BITTENCOURT, G., AND ROISENBERG, M. Estratégia para o controle dos robôs eyebot do ufsc-team: categoria small size do futebol de robôs. *XXVI Congresso da Sociedade Brasileira de Computação, III Encontro de Robótica Inteligente*, 117–125. Porto Alegre - Brasil, 2006.
- [61] STIVANELLO, M. E., LEAL, E. S., PALLUAT, N., AND STEMMER, M. R. Correspondência densa para sistemas de visão estereoscópica para robótica móvel. *Congresso Brasileiro de Automática*. Juiz de Fora - Brasil, 2008.

- [62] STIVANELLO, M. E., LEAL, E. S., PALLUAT, N., AND STEMMER, M. R. Desenvolvimento de uma biblioteca para sistemas de visão estereoscópica para robótica móvel. *VIII Conferência Internacional de Aplicações Industriais*. Poços de Caldas - Brasil, 2008.
- [63] TAYLOR, T., S.GEVA, AND BOLES, W. W. Monocular vision as a range sensor. *International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA)*. Gold Coast - Australia, 2004.
- [64] THOMPSON, S., AND KAGAMI, S. Humanoid robot localisation using stereo vision. *Proceedings of the 5th IEEE-RAS International Conference on Humanoid Robots*. Piscataway - USA, 2005.
- [65] THRUN, S., BURGARD, W., AND FOX, D. *Probabilistic Robotics*. The MIT Press. Cambridge - USA, 2006.
- [66] TIJTGAT, P., MAZYN, L., LAEY, C. D., AND LENOIR, M. The contribution of stereo vision to the control of braking. *Accident Analysis and Prevention*. Maryland Heights - USA, 2007.
- [67] TOMBARI, F., MATTOCCIA, S., AND STEFANO, L. D. Segmentation-based adaptive support for accurate stereo correspondence. *IEEE Pacific-Rim Symposium on Image and Video Technology*. Santiago, Chile, 2007.
- [68] TOULMINET, G., BERTOZZI, M., MOUSSET, S., BENSRAIR, A., AND BROGGI, A. Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis. *IEEE Transactions on image processing* 15, 8. Los Alamos - USA, 2006.
- [69] TURK, M. A., AND PENTLAND, A. P. Face recognition using eigenfaces. *Computer Vision and Pattern Recognition*, 586–591. Los Alamos - USA, 1991.
- [70] VALAVANIS, K. P., Ed. *IEEE Robotics & Automation Magazine - Finding The Right Path*, vol. 15. Institute of Electrical and Electronics Engineers, Piscataway-EUA, June 2008.

- 
- [71] WOODFILL, J. I., BUCK, R., JURASEK, D., GORDON, G., AND BROWN, T. 3D vision: Developing an embedded stereo-vision system. *Computer*. Los Alamos - May 2007.
  - [72] XU, M., ZHU, W., AND ZOU, Y. Research on reconfigurable robot controller based on arm and fpga. *IEEE International Symposium on Embedded Computing 0*, 216–222. Daejeon, 2008.
  - [73] YANG, R., AND POLLEFEYS, M. Multi-resolution real-time stereo on commodity graphics hardware. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1*, 211. Piscataway - USA, 2003.